VISIT…

A DSP A-Z

http://www.unex.ucla.edu

Digital Signal Processing An "A" to "Z"

R.W. Stewart
Signal Processing Division
Dept. of Electronic and Electrical Eng.
University of Strathclyde
Glasgow G1 1XW,
UK

Tel: +44 (0) 141 548 2396
Fax: +44 (0) 141 552 2487
E-mail: r.stewart@eee.strath.ac.uk

M.W. Hoffman
Department of Electrical Eng.
209N Walter Scott Eng. Center
PO Box 880511
Lincoln, NE 68588 0511
USA

Tel: +1 402 472 1979
Fax: +1 402 472 4732
Email:hoffman@unlinfo.unl.edu

© BlueBox Multimedia, R.W. Stewart 1998

The
DSPedia

An A-Z of Digital Signal Processing

This text aims to present relevant, accurate and readable definitions of common and not so common terms, algorithms, techniques and information related to DSP technology and applications. It is hoped that the information presented will complement the formal teachings of the many excellent DSP textbooks available and bridge the gaps that often exist between advanced DSP texts and introductory DSP.

While some of the entries are particularly detailed, most often in cases where the concept, application or term is particularly important in DSP, you will find that other terms are short, and perhaps even dismissive when it is considered that the term is not directly relevant to DSP or would not benefit from an extensive description.

There are 4 key sections to the text:

- DSP terms A-Z page 1
- Common Numbers associated with DSP page 427
- Acronyms page 435
- References page 443

Any comment on this text is welcome, and the authors can be emailed at r.stewart@.eee.strath.ac.uk, or hoffman@ee.unl.edu.

Bob Stewart, Mike Hoffman

1998

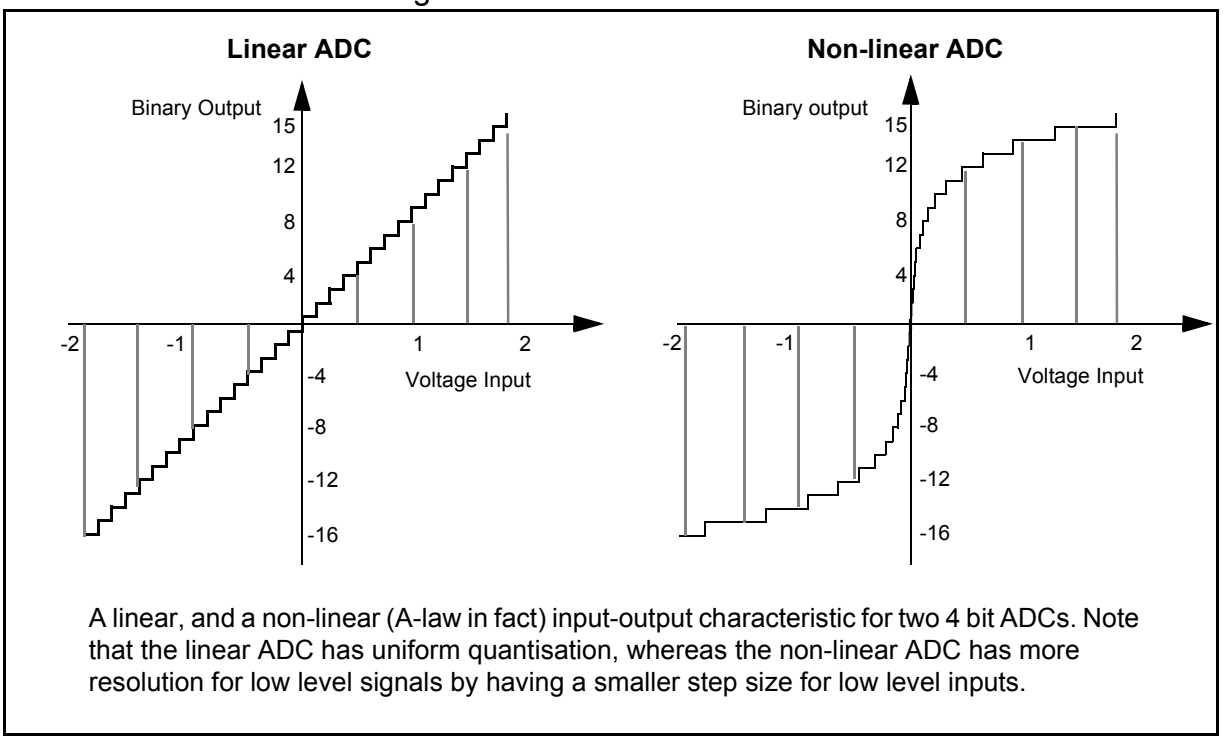
Published by BlueBox Multimedia.

A

A-series Recommendations: Recommendations from the International Telecommunication Union (ITU) telecommunications committee (ITU-T) outlining the work of the committee. See also *International Telecommunication Union, ITU-T Recommendations.*

A-law Compander: A defined standard nonlinear (logarithmic in fact) quantiser characteristic useful for certain signals. Non-linear quantisers are used in situations where a signal has a large dynamic range, but where signal amplitudes are more logarithmically distributed than they are linear. This is the case for normal speech.

Speech signals have a very wide dynamic range: Harsh “oh” and “b” type sounds have a large amplitude, whereas softer sounds such as “sh” have small amplitudes. If a uniform quantization scheme were used then although the loud sounds would be represented adequately the quieter sounds may fall below the threshold of the LSB and therefore be quantized to zero and the information lost. Therefore non-linear quantizers are used such that the quantization level at low input levels is much smaller than for higher level signals. To some extent this also exploits the logarithmic nature of human hearing.



A linear, and a non-linear (A-law in fact) input-output characteristic for two 4 bit ADCs. Note that the linear ADC has uniform quantisation, whereas the non-linear ADC has more resolution for low level signals by having a smaller step size for low level inputs.

A-law quantizers are often implemented by using a nonlinear circuit followed by a uniform quantizer. Two schemes are widely in use, the μ*-law* in the USA:

$$|y| = \frac{\ln(1+\mu|x|)}{\ln(1+\mu)} \tag{1}$$

and the A-law in Europe and Japan:

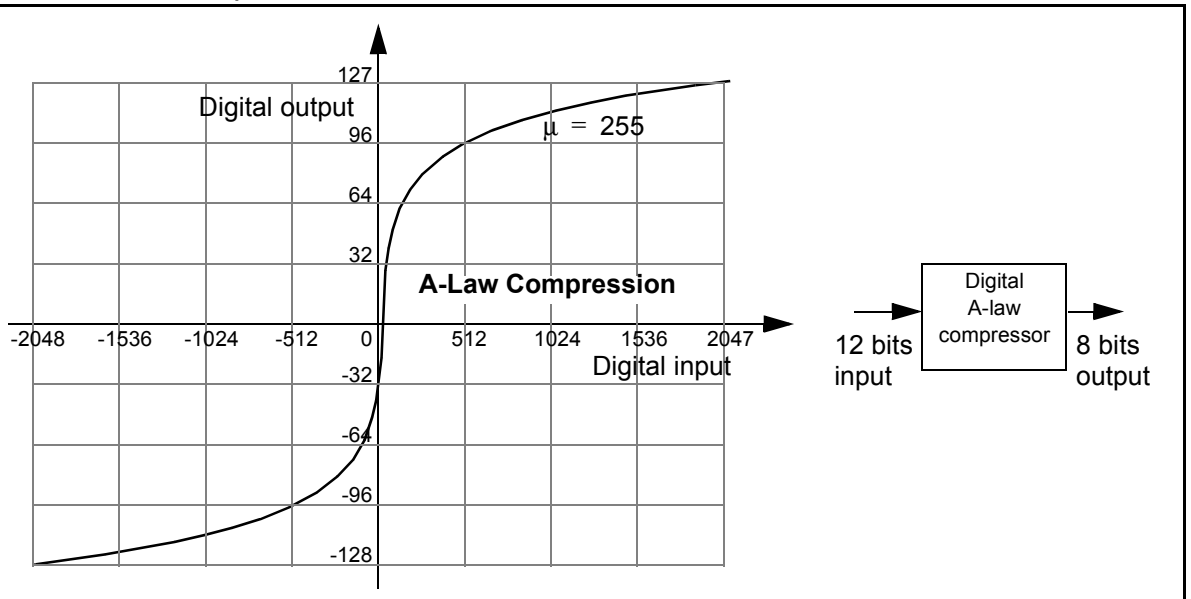
$$|y| = \frac{1+\ln A|x|}{1+\ln A} \tag{2}$$

where “ln” is the natural logarithm (base *e*), and the input signal x is in the range 0 to 1. The ITU have defined standards (G.711) for these quantisers where $\mu = 255$ and $A = 87.56$. The input/output characterisitcs of Eqs. 1 and 2 for these two values are virtually identical.

Although a non-linear quantiser can be produced with analogue circuitry, it is more usual that a linear quantiser will be used, followed by a digital implementation of the compressor. For example, if a signal has been digitised by a 12 bit linear ADC, then digital μ-law compression can be performed to compress to 8 bits using a modified version of Eq. 2:

$$|y| = 2^7 \frac{\ln(1+\mu|x/2^{11}|)}{\ln(1+\mu)} = 128 \frac{\ln(1+\mu|x/2048|)}{\ln(1+\mu)} \qquad (3)$$

where y is rounded to the nearest integer. After a signal has been compressed and transmitted, at the receiver it can be expanded back to its linear form by using an expander with the inverse characteristic to the compressor.



The ITU μ-law characteristic for compression from 12 bits to 8 bits. Note that if a value of $\mu = 0$ was used then the characteristic is linear, and for $\mu \to \infty$ the characteristic tends to a sigmoid/step function.

Listening tests for μ-law encoded speech reveal that compressing a linear resolution 12 bit speech signal (sampled at 8 kHz) to 8 bits, and then expanding back to a linearly quantised 12 bit signal does *not* degrade the speech quality to any significant degree. This can be quantitatively shown by considering the actual quantisation noise signals for the compressed and uncompressed speech signals.

In practice the use of DSP routines to perform Eq. 3 is not performed and a piecewise linear approximation (defined in G.711) to the μ- or A-law characteristic is used. See also *Companders, Compression,G-series Recommendations, m-law*.

Absolute Error: Consider the following example, if an analogue voltage of exactly v = 6.285 volts is represented to only one decimal place by rounding then $v' = 6.3$, and the *absolute error*, Δv, is defined as the difference between the true value and the estimated value. Therefore,

$$v = v' + \Delta v \qquad (4)$$

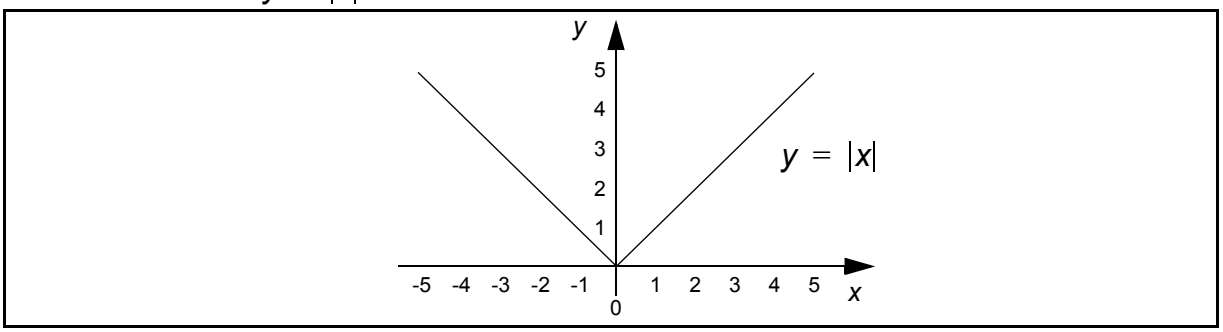
and

$$\Delta v = v - v' \tag{5}$$

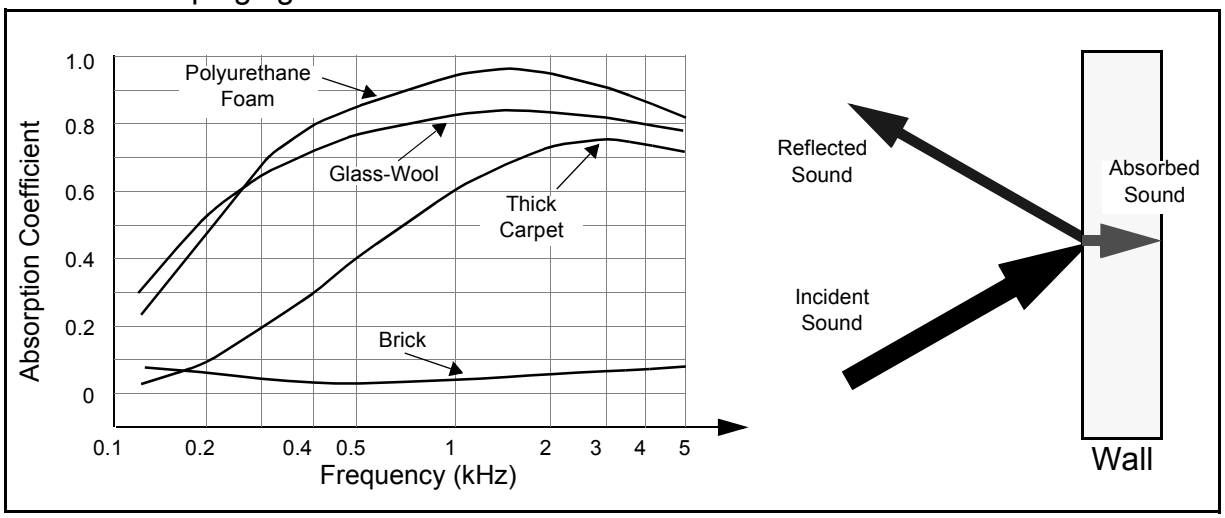
For this case Δv = -0.015 volts. Notice that *absolute error* does not refer to a positive valued error, but only that no normalization of the error has occurred. See also *Error Analysis, Quantization Error, Relative Error*.

Absolute Pitch: See entry for *Perfect Pitch*.

Absolute Value: The *absolute value* of a quantity, x, is usually denoted as $|x|$. If $x \geq 0$, then $|x| = x$, and if $x < 0$ then $|x| = -x$. For example $|12123| = 12123$, and $|-234.5| = 234.5$. The *absolute value* function $y = |x|$ is non-linear and is non-differentiable at $x = 0$.



Absorption Coefficient: When sound is absorbed by materials such as walls, foam etc., the amount of sound energy absorbed can be predicted by the material's *absorption coefficient* at a particular frequency. The absorption coefficients for a few materials are shown below. A 1.0 indicates that all sound energy is absorbed, and a 0, that none is absorbed. Sound that is not absorbed is reflected. The amplitude of reflected sound waves is given by $\sqrt{1-A}$ times the amplitude of the impinging sound wave.

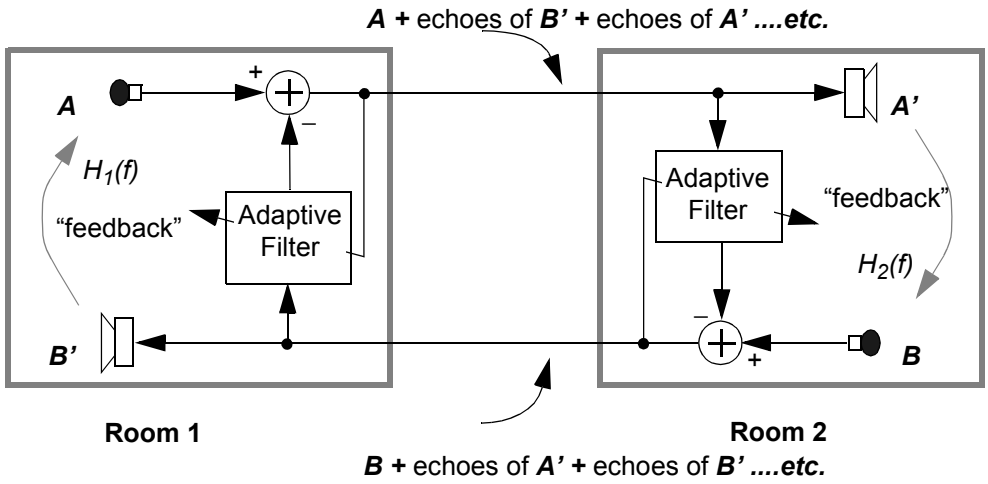


Accelerometer: A sensor that measures acceleration, often used for vibration sensing and attitude control applications.

Accumulator: Part of a DSP processor which can add two binary numbers together. The accumulator is part of the ALU (arithmetic logic unit). See also *DSP Processor*.

Accuracy: The *accuracy* of DSP system refers to the error of a quantity compared to its true value. See also *Absolute Error, Relative Error, Quantization Noise*.

Acoustic Echo Cancellation: For teleconferencing applications or hands free telephony, the loudspeaker and microphone set up in both locations causes a direct feedback path which can cause instability and therefore failure of the system. To compensate for this echo acoustic echo cancellers can be introduced:



When speaker A in room 1 speaks into microphone 1, the speech will appear at loudspeaker 2 in room 2. However the speech from loudspeaker 2 will be picked up by microphone 2, and transmitted back into room 1 via loudspeaker 1, which in turn is picked up by loudspeaker 1, and so on. Hence unless the loudspeaker and microphones in each room are acoustically isolated (which would require headphones), there is a direct feedback path which may cause stability problems and hence failure of the full duplex speakerphone. Setting up an adaptive filter at each end will attempt to cancel the echo at each outgoing line. Amplifiers, ADCs, DACs, communication channels etc. have been omitted to allow the problem to be clearly defined.

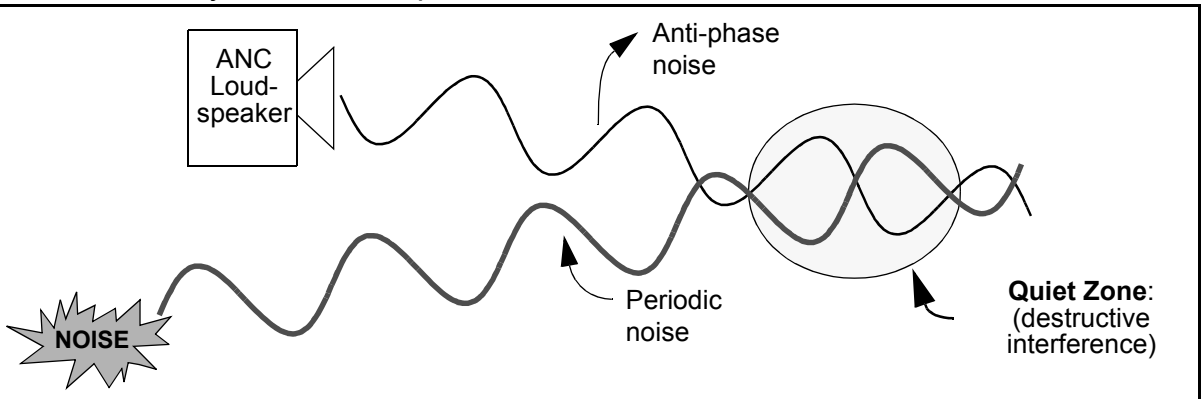
Teleconferencing is very dependent on adaptive signal processing strategies for acoustic echo control. Typically teleconferencing will sample at 8 or 16 kHz and the length of the adaptive filters could be thousands of weights (or coefficients), depending on the acoustic environments where they are being used. See also *Adaptive Signal Processing, Echo Cancellation, Least Mean Squares Algorithm, Noise Cancellation, Recursive Least Squares.*

Acoustics: The science of sound. See also *Absorption, Audio, Echo, Reverberation*.

Actuator: Devices which take electrical energy and convert it into some other form, e.g. loudspeakers, AC motors, Light emitting diodes (LEDs).

Active Filter: An analog filter that includes amplification components such as op-amps is termed an *active filter*; a filter that only has resistive, capacitive and inductive elements is termed a passive filter. In DSP systems analog filters are widely used for anti-alias and reconstruction filters, where *good* roll-off characteristics above $f_s/2$ are required. A simple RC circuit forms a first order (single pole) passive filter with roll of 20dB/decade (or 6dB/ocatve). By cascading RC circuits with an (active) buffer amplifier circuit, higher order filters (with more than one pole) can be easily designed. See also *Anti-alias Filter, Filters (Butterworth, Chebyshev, Bessel etc.) , Knee, Reconstruction Filter , RC Circuit, Roll-off*.

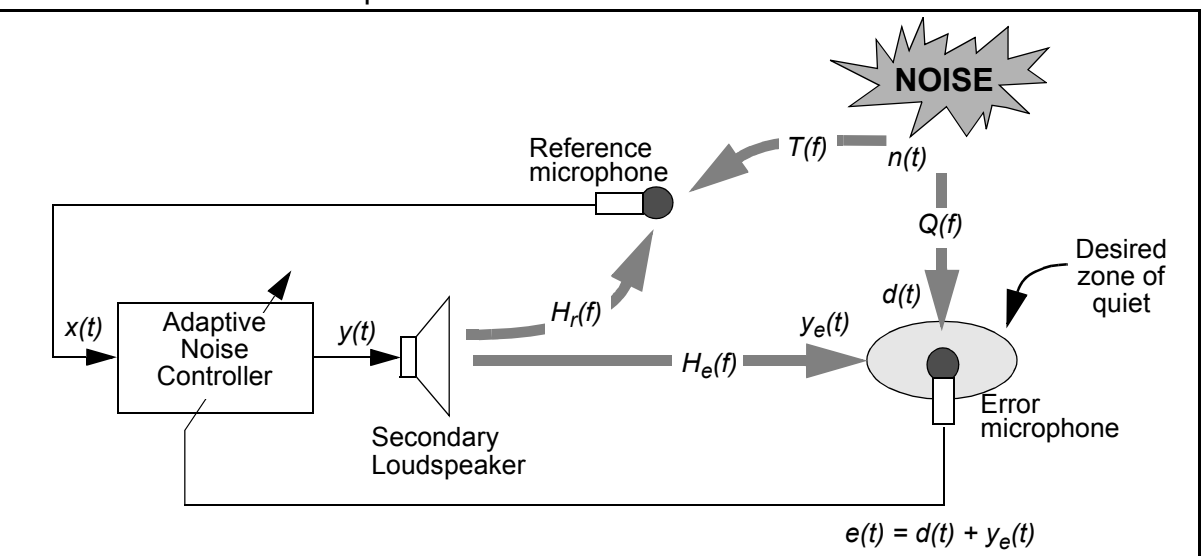
Active Noise Control (ANC): By introducing anti-phase acoustic waveforms, zones of quiet can be introduced at specified areas in space caused by the destructive interference of the offending noise and an artificially induced anti-phase noise:



The simple principle of active noise control.

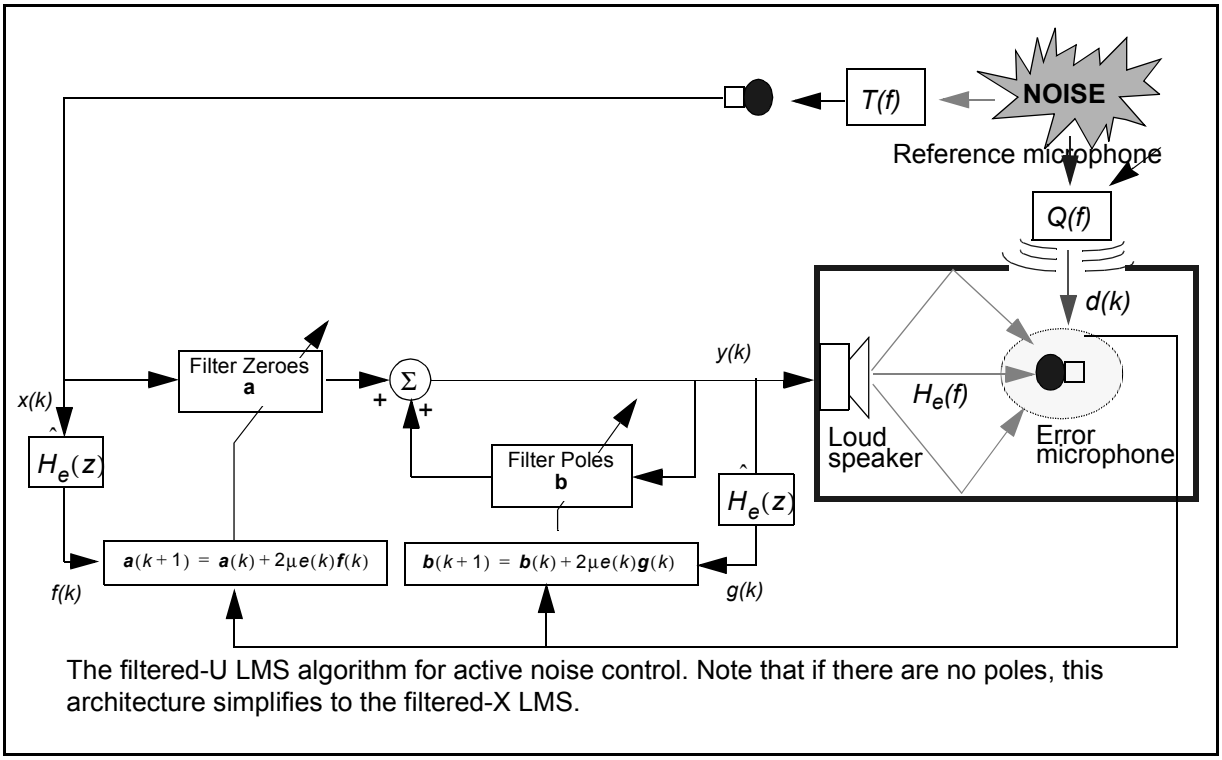
ANC works best for low frequencies up to around 600Hz. This can be intuitively argued by the fact that the wavelength of low frequencies is very long and it is easier to match peaks and troughs to create relatively large zones of quiet. Current applications for *ANC* can be found inside aircraft, in automobiles, in noisy industrial environments, in ventilation ducts, and in medical MRI equipment. Future applications include mobile telephones and maybe even noisy neighbors!

The general *active noise control* problem is:



The general set up of an active noise controller as a feedback loop where the aim is to minimize the error signal power.

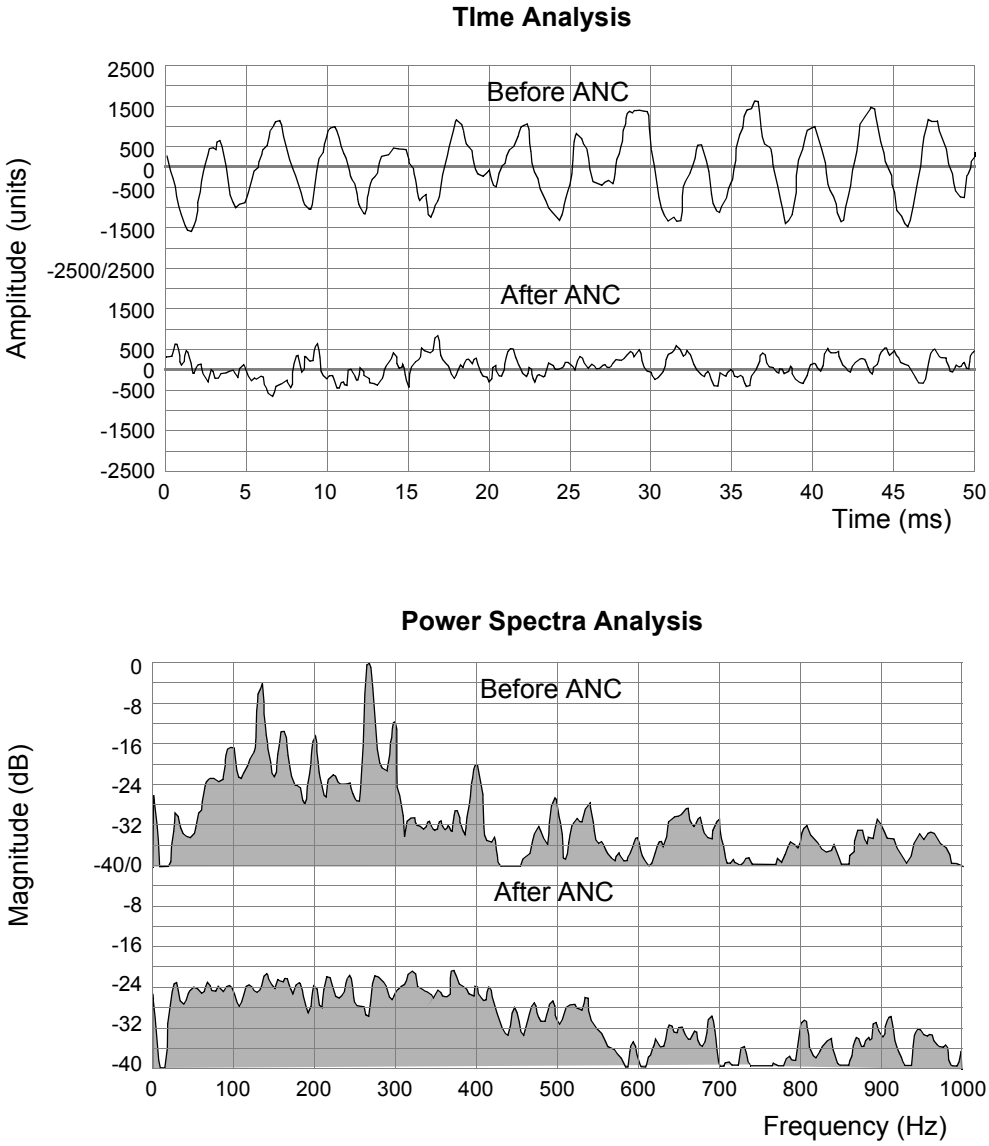
To implement an *ANC* system in real time the filtered-X LMS or filtered-U LMS algorithms can be used [68], [69]:



The filtered-U LMS algorithm for active noise control. Note that if there are no poles, this architecture simplifies to the filtered-X LMS.

The figure below shows the time and frequency domains for the *ANC* of an air conditioning duct. Note that the signals shown are represent the sound pressure level at the error microphone. In

general the zone of quiet does not extend much greater than $\lambda/4$ around the error microphone (where λ is the noise wavelength):



ANC inside air conditioning duct. The sound pressure levels shown represent the noise at an error microphone before and after switching on the noise canceller. The noise canceller clearly reduces the low frequency (periodic) noise components.

Sampling rates for *ANC* can be as low as 1kHz if the offending noise is very low in frequency (say 50-400Hz) but can be as high as 50 kHz for certain types of *ANC* headphones where very rapid adaption is required, even although the maximum frequency being cancelled is not more than a few kHz which would make the Nyquist rate considerably lower. See also *Active Vibration Control, Adaptive Line Enhancer, Adaptive Signal Processing, Least Mean Squares Algorithm, Least Mean Squares Filtered-X Algorithm Convergence, Noise Cancellation.*

Active Vibration Control (AVT): DSP techniques for AVT are similar to active noise cancellation (ANC) algorithms and architectures. Actuators are employed to introduce anti-phase vibrations in an attempt to reduce the vibrations of a mechanical system. See also *Active Noise Cancellation.*

AC-2: An Audio Compression algorithm developed by Dolby Labs and intended for applications such as high quality digital audio broadcasting. AC-2 claims compression ratios of 6:1 with sound quality almost indistinguishable from CD quality sound under almost all listening conditions. AC-2 is based on psychoacoustic modelling of human hearing. See also *Compression, Precision Adaptive Subband Coding (PASC)*.

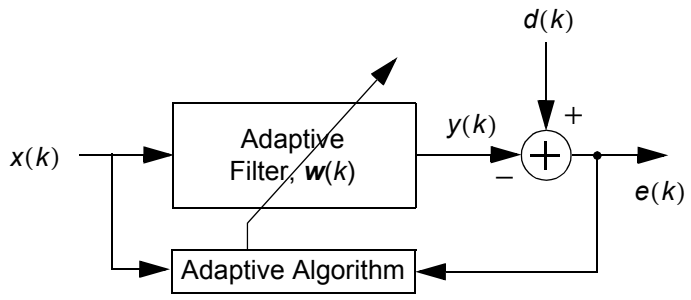
Adaptation: *Adaptation* is the auditory effect whereby a constant and noisy signal is perceived to become less loud or noticeable after prolonged exposure. An example would be the *adaptation* to the engine noise in a (loud!) propeller aircraft. See also *Audiology, Habituation, Psychoacoustics.*

Adaptive Differential Pulse Code Modulation (ADPCM): ADPCM is a family of speech compression and decompression algorithms which use adaptive quantizers and adaptive predictors to compress data (usually speech) for transmission. The CCITT standard of ADPCM allows an analog voice conversation sampled at 8kHz to be carried within a 32kbits/second digital channel . Three or four bits are used to describe each sample which represent the difference between two adjacent samples. See also *Differential Pulse Code Modulation (ADPCM), Delta Modulation, Continuously Variable Slope Delta Modulation (CVSD), G.721*.

Adaptive Beamformer: A spatial filter (beamformer) that has time-varying, data dependent (i.e., *adaptive*) weights. See also *Beamforming*.

Adaptive Equalisation: If the effects of a signal being passed through a particular system are to be "removed" then this is equalisation. See *Equalisation*.

Adaptive Filter: The generic *adaptive filter* can be represented as:



$$y(k) = \text{Filter}\{x(k), \mathbf{w}(k)\}$$

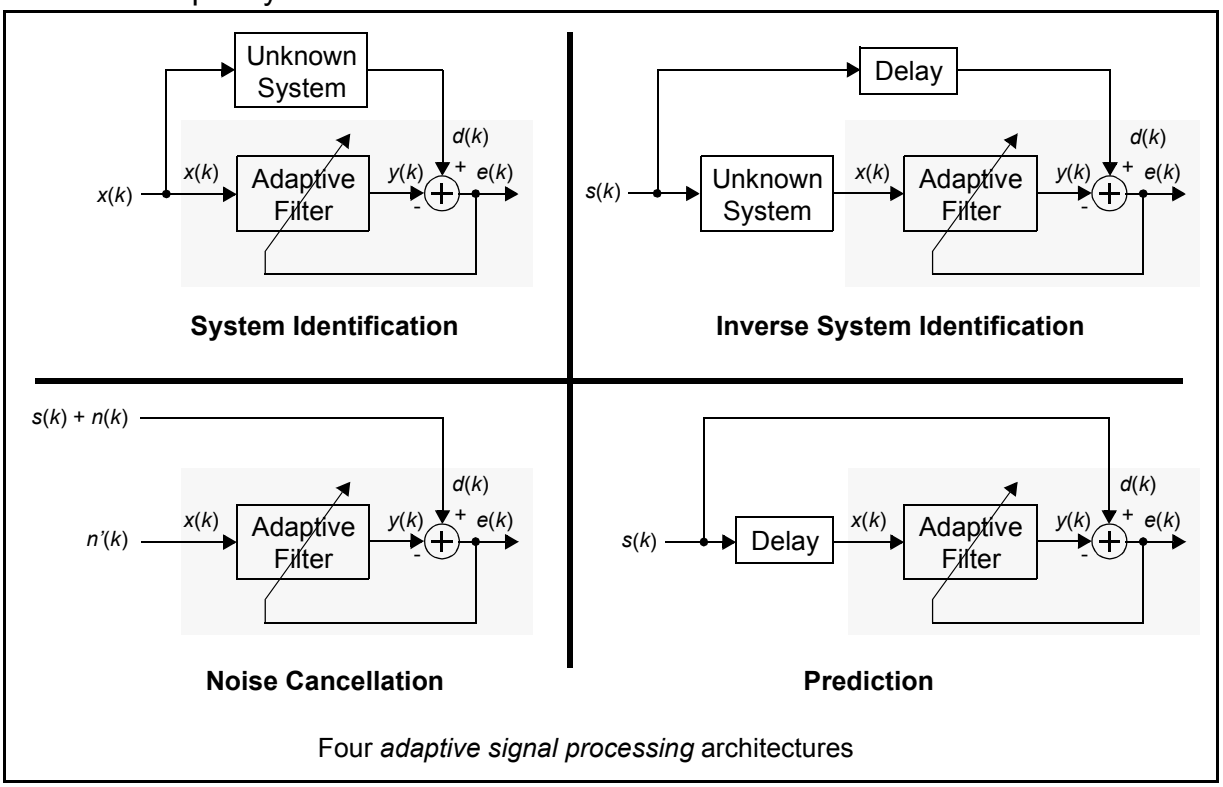
$$\mathbf{w}(k+1) = \mathbf{w}(k) + e(k)f\{d((k), x(k))\}$$

In the generic adaptive filter architecture the aim can intuitively be described as being to adapt the impulse response of the digital filter such that the input signal $x(k)$ is filtered to produce $y(k)$ which when subtracted from desired signal $d(k)$, will minimize the power of the error signal $e(k)$.

The adaptive filter output $y(k)$ is produced by the filter weight vector, $\mathbf{w}(k)$, convolved (in the linear case) with $x(k)$. The adaptive filter weight vector is updated based on a function of the error signal $e(k)$ at each time step k to produce a new weight vector, $\mathbf{w}(k+1)$ to be used at the next time step. This adaptive algorithm is used in order that the input signal of the filter, $x(k)$, is filtered to produce an output, $y(k)$, which is *similar* to the desired signal, $d(k)$, such that the power of the error signal, $e(k) = d(k) - y(k)$, is minimized. This minimization is essentially achieved by exploiting the correlation that should exist between $d(k)$ and $y(k)$.

The adaptive digital filter can be an FIR, IIR, Lattice or even a non-linear (Volterra) filter, depending on the application. The most common by far is the FIR. The adaptive algorithm can be based on gradient techniques such as the LMS, or on recursive least squares techniques such as the RLS. In general different algorithms have different attributes in terms of minimum error achievable, convergence time, and stability.

There are at least four general architectures that can be set up for *adaptive filters*: (1) System identification; (2) Inverse system identification; (3) Noise cancellation; (4) Prediction. Note that all of these architectures have the same generic adaptive filter as shown below (the "Adaptive Algorithm" block explicitly drawn above has been left out for illustrative convenience and clarity):



Four *adaptive signal processing* architectures

Consider first the system identification; at an intuitive level, if the adaptive algorithm is indeed successful at minimizing the error to zero, then by simple inspection the transfer function of the "Unknown System" must be identical to the transfer function of the adaptive filter. Given that the error of the adaptive filter is now zero, then the adaptive filters weights are no longer updated and will remain in a steady state. As long as the unknown system does not change its characteristics we have now successfully identified (or modelled) the system. If the adaption was not perfect and the error is "very small" rather than zero (which is more likely in real applications) then it is fair to say the we have a good model rather than a perfect model.

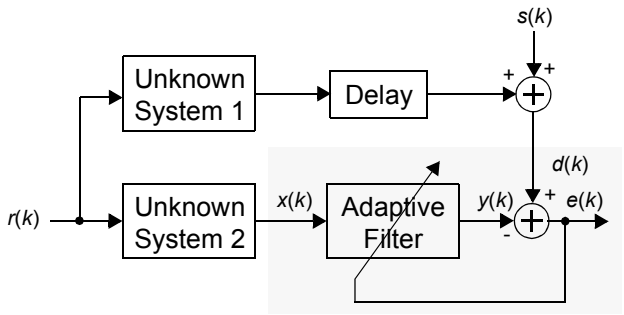
Similarly for the inverse system identification if the error adapts to zero over a period of time, then by observation the transfer function of the adaptive filter must be the exact inverse of the "Unknown System". (Note that the "Delay" is necessary to ensure that the problem is causal and therefore solvable with real systems, i.e. given that the "Unknown System" may introduce a time delay in producing $x(k)$, then if the "Delay" was not present in the path to the desired signal the system would be required produced an anti-delay or look ahead in time - clearly this is impossible.)

For the noise cancellation architecture, if the input signal is $s(k)$ which is corrupted by additive noise, $n(k)$, then the aim is to use a correlated noise reference signal, $n'(k)$ as an input to the

adaptive filter, such that when performing the adaption there is only information available to implicitly model the noise signal, $n(k)$ and therefore when this filter adapts to a steady state we would expect that $e(k) \approx s(k)$.

Finally, for the prediction filter, if the error is set to be adapted to zero, then the adaptive filter must predict future elements of the input $s(k)$ based only on past observations. This can be performed if the signal $s(k)$ is periodic and the filter is long enough to "remember" past values. One application therefore of the prediction architecture could be to extract periodic signals from stochastic noise signals. The prediction filter can be extended to a "smoothing filter" if data are processed off-line -- this means that samples before and after the present sample are filtered to obtain an estimate of the present sample. Smoothing cannot be done in real-time, however there are important applications where real-time processing is not required (e.g., geophysical seismic signal processing).

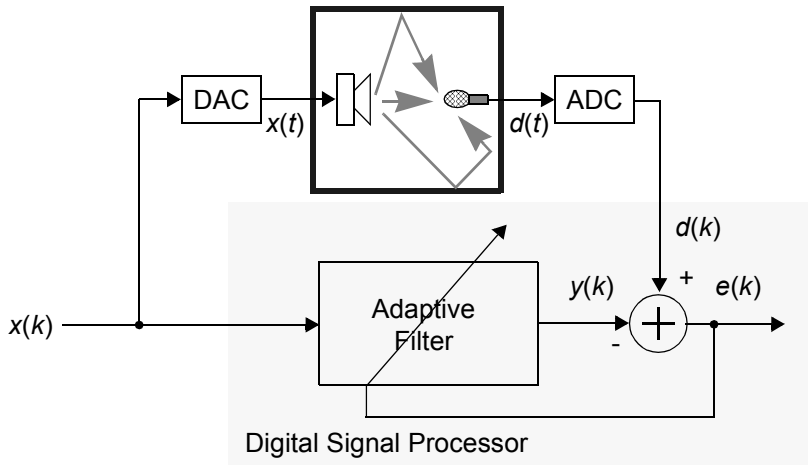
A particular application may have elements of more than one single architecture, for example in the following, if the adaptive filter is successful in modelling "Unknown System 1", and inverse modelling "Unknown System 2", then if $s(k)$ is uncorrelated with $r(k)$ then the error signal is likely to be $e(k) \approx s(k)$:



An *adaptive filtering* architecture incorporating elements of system identification, inverse system identification and noise cancellation

In the four general architectures shown above the unknown systems being investigated will normally be analog in nature, and therefore suitable ADCs and DACs would be used at the various

analog input and output points as appropriate. For example if an adaptive filter was being used to find a model of a small acoustic enclosure the overall hardware set up would be:

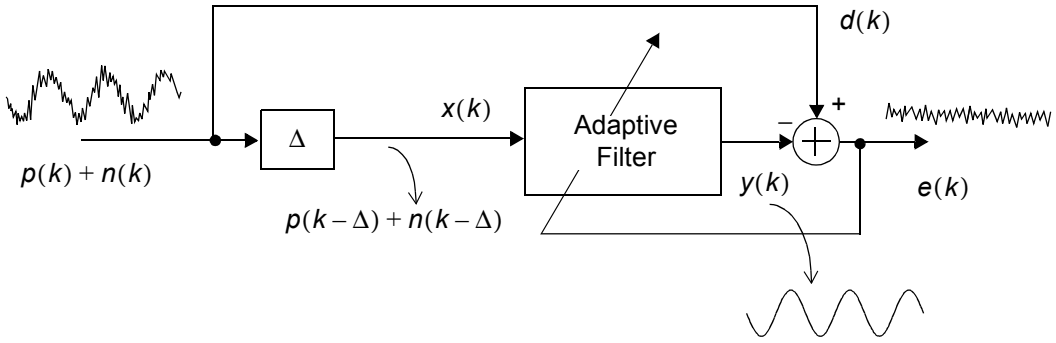


The analog-digital interfacing for a system identification, or modelling, of an acoustic transfer path using a loudspeaker and microphone.

See also *Adaptive Signal Processing, Acoustic Echo Cancellation, Active Noise Control, Adaptive Line Enhancer, Echo Cancellation, Least Mean Squares (LMS) Algorithm, Least Squares, Noise Cancellation, Recursive Least Squares, Wiener-Hopf Equations.*

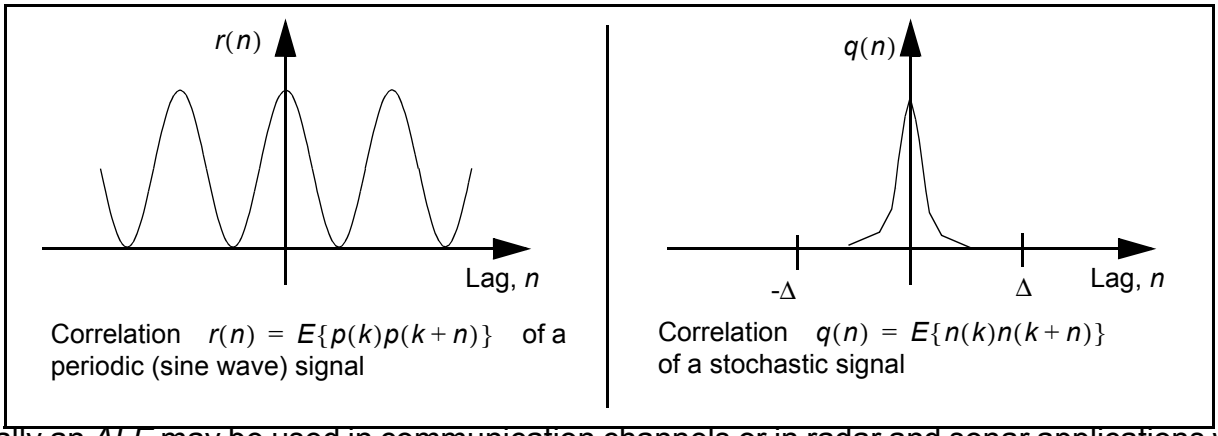
Adaptive Infinite Impulse Response (IIR) Filters: See *Least Mean Squares IIR Algorithms.*

Adaptive Line Enhancer (ALE): An adaptive signal processing structure that is designed to enhance or extract periodic (or predictable) components:



An *adaptive line enhancer*. The input signal consists of a periodic component, $p(k)$ and a stochastic component, $n(k)$. The delay, Δ, is long enough such that the stochastic component at the input to the adaptive filter, $n(k-\Delta)$ is decorrelated with the input $n(k)$. For periodic signal the delay does not decorrelate $p(k)$ and $p(k-\Delta)$. When the adaptive filter adapts it will therefore only cancel the periodic signal.

The delay, Δ, should be long enough to decorrelate the broadband "noise-like" signal, resulting in an adaptive filter which extracts the narrowband periodic signal at filter output $y(k)$ (or removes the periodic noise from a wideband signal at $e(k)$). An *ALE* exploits the knowledge that the signal of interest is periodic, whereas the additive noise is stochastic. If the decorrelation delay, Δ, is long enough then the stochastic noise presented to the $d(k)$ input is uncorrelated with the noise presented to the $x(k)$ input, however the periodic noise remains correlated:



Typically an *ALE* may be used in communication channels or in radar and sonar applications where a low level sinusoid is masked by white or colored noise. In a telecommunications system, an ALE could be used to extract periodic DTMF signals from very high levels of stochastic noise. Alternatively note that the *ALE* can be used to extract the periodic noise from the stochastic signal by observing the signal $e(k)$. See also *Adaptive Signal Processing, Least Mean Squares Algorithm, Noise Cancellation.*

Adaptive Noise Cancellation: See *Adaptive Signal Processing*, *Noise Cancellation.*

Adaptive Signal Processing: The discrete mathematics of adaptive filtering, originally based on the least squares minimization theory of the celebrated 19th Century German mathematician Gauss. Least squares is of course widely used in statistical analysis and virtually every branch of science and engineering. For many DSP applications, however, least squares minimization is applied to real time data and therefore presents the challenge of producing a real time implementation to operate on data arriving at high data rates (from 1kHz to 100kHz), and with *loosely* known statistics and properties. In addition, other cost functions besides least squares are also used.

One of the first suggestions of adaptive DSP algorithms was in Widrow and Hoff's classic paper on the adaptive switching circuits and the least mean squares (LMS) algorithm at the IRE WESCON Conference in 1960. This paper stimulated great interest by providing a practical and potentially real time solution for least squares implementation. Widrow followed up this work with two definitive and classic papers on *adaptive signal processing* in the 1970s [152], [153].

Adaptive signal processing has found many applications. A generic breakdown of these applications can be made into the following categories of signal processing problems: signal detection (is it there?), signal estimation (what is it?), parameter or state estimation, signal compression, signal synthesis, signal classification, etc. The common attributes of *adaptive signal processing* applications include time varying (adaptive) computations (processing) using sensed input values (signals).See also *Acoustic Echo Cancellation, Active Noise Control, Adaptive Filter, Adaptive Line Enhancer, Echo Cancellation, Least Mean Squares (LMS) Algorithm, Least Squares, Noise Cancellation, Recursive Least Squares, Wiener-Hopf Equations.*

Adaptive Spectral Perceptual Entropy Coding (ASPEC): *ASPEC* is a means of providing psychoacoustic compression of hifidelity audio and was developed by AT&T Bell Labs, Thomson and the Fraunhofer society amongst others. In 1990 features of the ASPEC coding system were incorporated into the International Organization for Standards MPEG-1 standard ISO in combination with MUSICAM. See also *Masking Pattern Adapted Universal Subband Integrated*

Coding and Multiplexing (MUSICAM), Precision Adaptive Subband Coding (PASC), Spectral Masking, Psychoacoustics, Temporal Masking.

Adaptive Step Size: See *Step Size Parameter.*

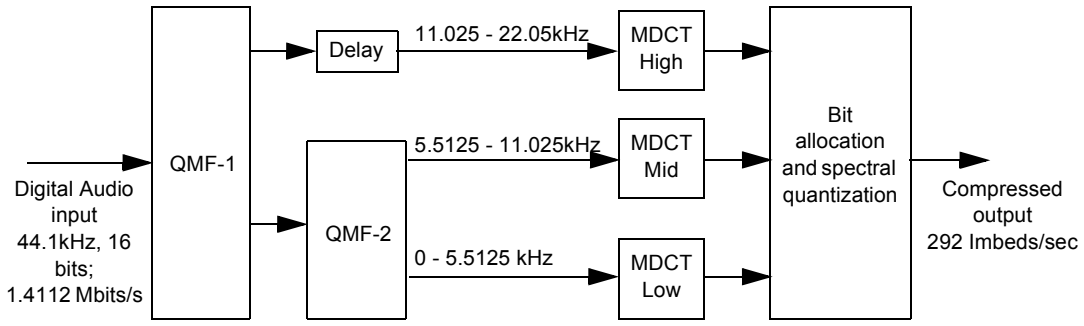
Adaptive Transform Acoustic Coding (ATRAC): *ATRAC* coding is used for compression of hifidelity audio (usually starting with 16 bit data at 44.1kHz) to reduce storage requirement on recording mediums such as the mini-disc (MD) [155]. *ATRAC* achieves a compression ratio of almost 5:1 with very little perceived difference to uncompressed PCM quality. ATRAC exploits psychoacoustic (spectral) masking properties of the human ear and effectively compresses data by varying the bit resolution used to code different parts of the audio spectrum. More information on the mini-disc (and also ATRAC) can be found in [155].

ATRAC has three key coding stages. First is the subband filtering which splits the signal into three subbands, (low:0 - 5.5 kHz; mid:5.5 - 11kHz; high:11- 22kHz) using a two stage quadrature mirror filter (QMF) bank.

The second stage them performs a modified discrete cosine transform (MDCT) to produce a frequency representation of the signal. The actual length (no. of samples) of the transform is controlled adaptively via an internal decision process and either uses time frame lengths of 11.6ms (when in long mode) for all frequency bands, and 1.45ms (when in short mode) for the high frequency band, and 2.9ms (also called short mode) for the low and mid frequency bands. The choice of mode is usually long, however if a signal has rapidly varying instantaneous power (when say a cymbal is struck) short mode may be required in the low and mid frequency bands to adequately code the rapid attack portion of the waveform.

Finally the third stage is to consider the spectral characteristics of the three subbands and allocate bit resolution such that spectral components below the threshold of hearing, are not encoded, and that the spectral masking attributes of the signal spectrum are exploited such that the number of bits required to code certain frequency bands is greatly reduced. (See entry for Precision Adaptive Subband Coding (PASC) for a description of quantization noise masking.) ATRAC splits the frequencies from the MDCT into a total of 52 frequency bins which are of varying bandwidth based on the width of the critical bands in the human auditory mechanism. *ATRAC* then compands and requantizes using a block floating point representation. The wordlength is determined by the bit

allocation process based on psychoacoustic models. Each input 11.6 ms time frame of 512 × 16 bit samples or 1024 bytes is compressed to 212 bytes (4.83:1 compression ratio).



The three stages of *adaptive transform acoustic coding* (*ATRAC*): (1) Quadrature mirror filter (QMF) subband coding; (2) Modified Discrete Cosine Transform (MDCT); (3) Bit allocation and spectral masking/quantization decision. Data is input for coding in time frames of 512 samples (1024 bytes) and compressed into 212 bytes.

ATRAC decoding from compressed format back to 44.1kHz PCM format is achieved by first performing an inverse MDCT on the three subbands (using long mode or short mode data lengths as specified in the coded data). The three time domain signals produced are then reconstructed back into a time domain signal using QMF synthesis filters for output to a DAC. See also *Compact Disc, Data Compression, Frequency Range of Hearing, MiniDisc (MD), Psychoacoustics, Precision Adaptive Subband Coding (PASC), Spectral Masking, Subband Filtering, Temporal Masking, Threshold of Hearing*.

Additive White Gaussian Noise: The most commonly assumed noise channel in the analysis and design of communications systems. Why is this so? Well, for one, this assumption allows analysis of the resulting system to be tractable (i.e., we can do the analysis). In addition, this is a very good model of electronic circuit noise. In communication systems the modulated signal is often so weak that this circuit noise becomes a dominant effect. The model of a flat (i.e., white) spectra is good in electronic circuits up to about 10^{12}Hz. See also *White Noise.*

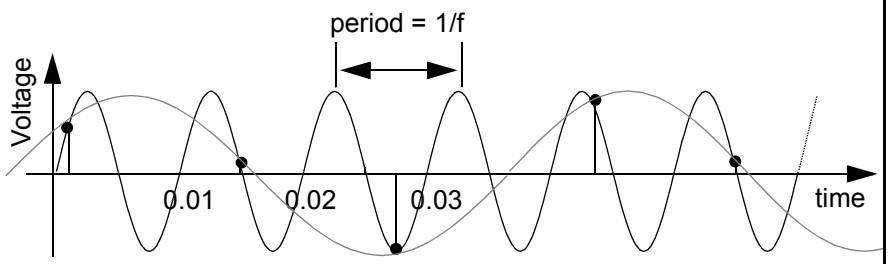
Address Bus: A collection of wires that are used for sending memory address information either inter-chip (between chips) or intra-chip (within a chip). Typically DSP address buses are 16 or 32 bits wide. See also *DSP Processor*.

Address Registers: Memory locations inside a DSP processor that are used as temporary storage space for addresses of data stored somewhere in memory. The *address register* width is always greater than or equal to (normally the same) the width of the DSP processor address bus. Most DSP processors have a number of address registers. See also *DSP Processor*.

AES/EBU: See *Audio Engineering Society, European Broadcast Union.*

Aliasing: An irrecoverable effect of sampling a signal too slowly. High frequency components of a signal (over one-half the sampling frequency) cannot be accurately reconstructed in a digital system. Intuitively, the problem of sampling too slowly (*aliasing*) can be understood by considering that rapidly varying signal fluctuations that take place in between samples cannot be represented at the output. The distortion created by sampling these high frequency signals too slowly is not

reversible and can only be avoided by proper *aliasing* protection as provided by an anti-alias filter or a an oversampled Analog to Digital converter.



Sampling a 100 Hz sine wave at only 80 Hz causes aliasing, and the output signal is interpreted as a 20 Hz sine wave, i.e.

See also *Anti-alias Filter, Oversampling*.

Algorithm: A mathematical based computational method which forms a set of well defined rules or equations for performing a particular task. For example, the FFT *algorithm* can be coded into a DSP processor assembly language and then used to calculate FFTs from stored (or real-time) digital data.

All-pass Filter: An *all-pass filter* passes all input frequencies with the same gain, although the phase of the signal will be modified. (A true *all-pass filter* has a gain of one.) *All-pass filters* are used for applications such as group delay equalisation, notch filtering design, Hilbert transform implementation, musical instruments synthesis [43] .

The simplest all pass filter is a simple delay! This “filter” passes all frequencies with the same gain, has linear phase response and introduces a group delay of one sample at all frequencies:

time domain	**z-domain**
$x(k) \rightarrow \boxed{\triangle} \rightarrow y(k)$	$Y(z) = z^{-1}H(z)$
$y(k) = x(k-1)$	$H(z) = \dfrac{Y(z)}{X(z)} = z^{-1}$

A simple *all pass filter*. All frequencies are passed with the same gain.

A more general representation of some types of *all pass filters* can be represented by the general z-domain transfer function for an infinite impulse response (IIR) *N* pole, *N* zero filter:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a_0^* z^{-N} + a_1^* z^{-N+1} + \ldots + a_{N-1}^* z^{-1} + a_N^*}{a_0 + a_1 z^{-1} + \ldots + a_{N-1} z^{-N+1} + a_N z^{-N}} = \frac{z^{-N} A^*(z^{-1})}{A(z)} \tag{6}$$

where a^* is the complex conjugate of a. Usually the filter weights are real, therefore $a = a^*$, and we set $a_0 = 1$:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-N} + a_1 z^{-N+1} + \ldots + a_{N-1} z^{-1} + a_N}{1 + a_1 z^{-1} + \ldots + a_{N-1} z^{-N+1} + a_N z^{-N}} = \frac{z^{-N} A(z^{-1})}{A(z)} \tag{7}$$

We can easily show that $|H(z)| = a_N$ (see below) for all frequencies. Note that the numerator polynomial $z^{-N}A(z)$ is simply the ordered reversed z-polynomial of the denominator $A(z)$. For an input signal $x(k)$ the discrete time output of an *all-pass filter* is:

$$\begin{aligned} y(k) = {} & a_N x(k) + a_{N-1}x(k-1) + \ldots + a_1 x(k-N+1) + x(k-N) + \\ & + a_1 y(k-1) + \ldots + a_{N-1}y(k+N-1) + a_N x(k-N) \end{aligned} \tag{8}$$

In order to be stable, the poles of the *all-pass filter* must lie within the unit circle. Therefore for the denominator polynomial, if the N roots of the polynomial $A(z)$ are:

$$A(z) = (1-p_1 z^{-1})(1-p_2 z^{-1})\ldots(1-p_N z^{-1}) \tag{9}$$

then $|p_n| < 1$ for $n = 1$ to N in order to ensure all poles are within the unit circle. The poles and zeroes of the all pass filter are therefore:

$$H(z) = \frac{a_N(1-p_1^{-1}z^{-1})(1-p_2^{-1}z^{-1})\ldots(1-p_N^{-1}z^{-1})}{(1-p_1 z^{-1})(1-p_2 z^{-1})\ldots(1-p_N z^{-1})} \tag{10}$$

where the roots of the zeroes polynomial $A(z^{-1})$ are easily calculated to be the inverse of the poles (see following example).

To illustrate the relationship between roots of z-domain polynomial and of its order reversed polynomial, consider a polynomial of order 3 with roots at $z = p_1$ and $z = p_2$:

$$\begin{aligned} & 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} = (1-p_1 z^{-1})(1-p_2 z^{-1})(1-p_3 z^{-1}) \\ & = 1 - (p_1+p_2+p_3)z^{-1} + (p_1 p_2 + p_2 p_3 + p_1 p_3)z^{-2} + p_1 p_2 p_3 z^{-3} \end{aligned}$$

Then replacing z with z^{-1} gives:

$$1 + a_1 z^{1} + a_2 z^{2} + a_3 z^{3} = (1-p_1 z)(1-p_2 z)(1-p_3 z)$$

and therefore multiplying both sides by z^{-3} gives:

$$\begin{aligned} z^{-3}(1 + a_1 z^{1} + a_2 z^{2} + a_3 z^{3}) &= z^{-3}(1-p_1 z)(1-p_2 z)(1-p_3 z) \\ z^{-3} + a_1 z^{-2} + a_2 z^{-1} + a_3 &= (z^{-1}-p_1)(z^{-1}-p_2)(z^{-1}-p_3) \\ &= -p_1 p_2 p_3 (1-p_1^{-1}z^{-1})(1-p_2^{-1}z^{-1})(1-p_3^{-1}z^{-1}) \\ &= -a_3 (1-p_1^{-1}z^{-1})(1-p_2^{-1}z^{-1})(1-p_3^{-1}z^{-1}) \end{aligned}$$

hence revealing the roots of the order reversed polynomial to be at $z = 1/p_1$, $z = 1/p_2$ and $z = 1/p_3$.

Of course, if all of the poles of Eq. 10 lie within the z-domain unit circle then all of the zeroes of the denominator of Eq. 10 will necessarily lie outside of the unit circle of the z-domain, i.e. when $|p_n| < 1$ for $n = 1$ to N then $|p_n^{-1}| > 1$ for $n = 1$ to N. Therefore an *all pass filter* is maximum phase.

The magnitude frequency response of the pole at $z = p_i$ and the zero at $z = p_i^{-1}$ is:

$$|H_i(e^{j\omega})| = \left.\frac{|1-p_i^{-1}z^{-1}|}{|1-p_i z^{-1}|}\right|_{z=e^{j\omega}} = \frac{1}{|p_i|} \tag{11}$$

If we let $p_i = x + jy$ then the frequency response is found by evaluating the transfer function at $z = e^{j\omega}$:

$$H_i(e^{j\omega}) = \frac{1 - p_i^{-1}e^{-j\omega}}{1 - p_i e^{-j\omega}} = \frac{1}{p_i}\left(\frac{p_i - e^{-j\omega}}{1 - p_i e^{-j\omega}}\right) = \frac{1}{p_i}G(e^{j\omega})$$

where $|G(e^{j\omega})| = 1$. This can be shown by first considering that:

$$G(e^{j\omega}) = = \frac{x + jy - (\cos\omega - j\sin\omega)}{1 - (x + jy)(\cos\omega - j\sin\omega)} = \frac{(x - \cos\omega) + j(y - \sin\omega)}{1 - x\cos\omega - y\sin\omega + j(x\sin\omega - y\cos\omega)}$$

and therefore the (squared) magnitude frequency response of $G(e^{j\omega})$ is:

$$|G(e^{j\omega})|^2 = \frac{(x - \cos\omega)^2 + (y - \sin\omega)^2}{(1 - (x\cos\omega + y\sin\omega))^2 + (x\sin\omega - y\cos\omega)^2}$$

$$= \frac{(x^2 - 2x\cos\omega + \cos^2\omega) + (y^2 - 2y\sin\omega + \sin^2\omega)}{1 - 2x\cos\omega - 2y\sin\omega + (x\cos\omega + y\sin\omega)^2 + x^2\sin^2\omega + y^2\cos^2\omega - 2xy\sin\omega\cos\omega}$$

$$= \frac{(\sin^2\omega + \cos^2\omega) + x^2 + y^2 - 2x\cos\omega - 2b\sin\omega}{1 + x^2(\sin^2\omega + \cos^2\omega) + y^2(\sin^2\omega + \cos^2\omega) - 2x\cos\omega + 2y\sin\omega}$$

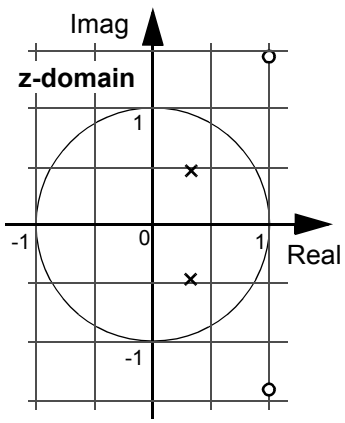
$$= \frac{1 + x^2 + y^2 - 2x\cos\omega - 2y\sin\omega}{1 + x^2 + y^2 - 2x\cos\omega + 2y\sin\omega} = 1$$

Hence: $|H_i(e^{j\omega})| = \frac{1}{|p_i|} = \frac{1}{\sqrt{x^2 + y^2}}$

Therefore the magnitude frequency response of the all pass filter in Eq. 10 is indeed “flat” and given by:

$$|H(e^{j\omega})| = a_N|H_1(e^{j\omega})||H_2(e^{j\omega})|\ldots|H_N(e^{j\omega})| = \frac{a_N}{|p_1||p_2|\ldots|p_N|} = 1 \qquad (12)$$

From Eq. 7 and 10 it is easy to show that $a_N = |p_1||p_2|\ldots|p_N|$.



Consider the poles and zeroes of a simple 2nd order all-pass filter transfer function (found by simply using the quadratic formula):

$$H(z) = \frac{1 + 2z^{-1} + 3z^{-2}}{3 + 2z^{-1} + z^{-2}}$$

$$= \frac{(1 - (1 + j\sqrt{2})z^{-1})(1 - (1 - j\sqrt{2})z^{-1})}{3(1 - (1/3 + j\sqrt{2}/3)z^{-1})(1 - (1/3 - j\sqrt{2}/3)z^{-1})}$$

$$= \frac{1}{|p_1||p_2|} \cdot \frac{(1 - p_1^{-1}z^{-1})(1 - p_2^{-1}z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})}$$

and obviously $p_1 = 1/3 - j\sqrt{2}/3$ and $p_2 = 1/3 + j\sqrt{2}/3$ and $p_1^{-1} = 1 - j\sqrt{2}$ and $p_2^{-1} = 1 + j\sqrt{2}$. This example demonsrates that given that the poles must be inside the unit circle for a stable filter, the zeroes will always be outside of the unit circle, i.e. maximum phase.

Any non-minimum phase system (i.e. zeroes outside the unit circle) can always be described as a cascade of a minimum phase filter and a maximum phase all-pass filter. Consider the non-minimum phase filter:

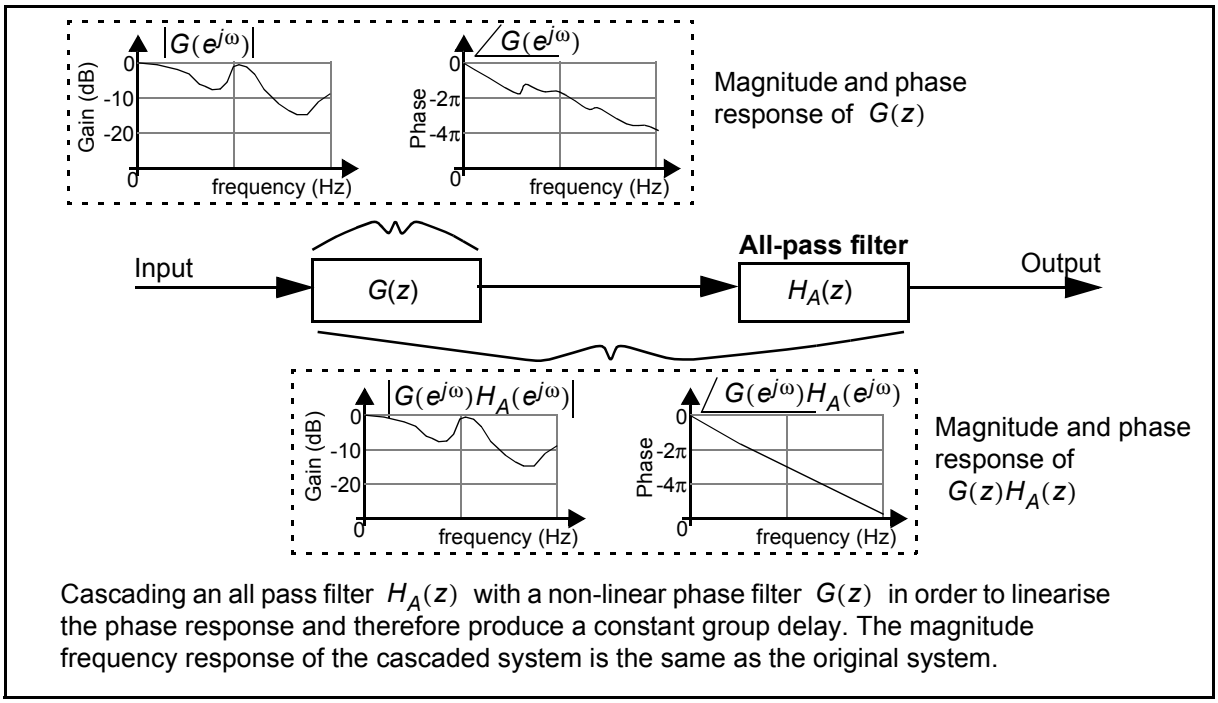
$$H(z) = \frac{(1-\alpha_1 z^{-1})(1-\alpha_2 z^{-1})(1-\alpha_3 z^{-1})(1-\alpha_4 z^{-1})}{(1-\beta_1 z^{-1})(1-\beta_2 z^{-1})(1-\beta_3 z^{-1})} \tag{13}$$

where the poles, $\beta_1, \beta_2,$ and β_3 are inside the unit circle (to ensure a stable filter) and the zeroes α_1 and α_2 are inside the unit circle, but the zeroes α_3 and α_4 are outside of the unit circle. This filter can be written in the form of a minimum phase system cascaded with an all-pass filter by rewriting as:

$$\begin{aligned} H(z) &= \left(\frac{(1-\alpha_1 z^{-1})(1-\alpha_2 z^{-1})(1-\alpha_3 z^{-1})(1-\alpha_4 z^{-1})}{(1-\beta_1 z^{-1})(1-\beta_2 z^{-1})(1-\beta_3 z^{-1})}\right)\left(\frac{(1-\alpha_3^{-1} z^{-1})(1-\alpha_4^{-1} z^{-1})}{(1-\alpha_3^{-1} z^{-1})(1-\alpha_4^{-1} z^{-1})}\right) \\ &= \underbrace{\left(\frac{(1-\alpha_1 z^{-1})(1-\alpha_2 z^{-1})(1-\alpha_3^{-1} z^{-1})(1-\alpha_4^{-1} z^{-1})}{(1-\beta_1 z^{-1})(1-\beta_2 z^{-1})(1-\beta_3 z^{-1})}\right)}_{\text{Minimum phase filter}}\underbrace{\left(\frac{((1-\alpha_3 z^{-1})(1-\alpha_4 z^{-1}))}{(1-\alpha_3^{-1} z^{-1})(1-\alpha_4^{-1} z^{-1})}\right)}_{\text{All-pass maximum phase filter}} \end{aligned} \tag{14}$$

Therefore the minimum phase filter has zeroes inside the unit circle at $z = \alpha_3^{-1}$, $z = \alpha_4^{-1}$ and has exactly the same magnitude frequency response as the original filter and the gain of the all-pass filter being 1. See also *All-pass Filter-Phase Compensation, Digital Filter, Infinite Impulse Response Filter, Notch Filter*.

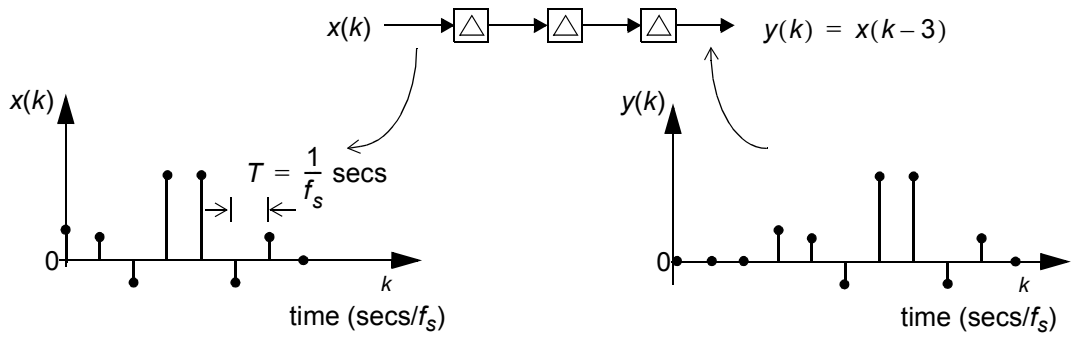
All-pass Filter, Phase Compensation: *All pass filters* are often used for phase compensation or group delay equalisation where the aim is to cascade an all-pass filter with a particular filter in order to achieve a linear phase response in the passband and leave the magnitude frequency response unchanged. (Given that signal information in the stopband is unwanted then there is usually no need to phase compensate there!). Therefore if a particular filter has a non-linear phase response and therefore non-constant group delay, then it may be possible to design a phase compensating *all-pass filter*:



Cascading an all pass filter $H_A(z)$ with a non-linear phase filter $G(z)$ in order to linearise the phase response and therefore produce a constant group delay. The magnitude frequency response of the cascaded system is the same as the original system.

See also *Digital Filter, Infinite Impulse Response Filter, Notch Filter*.

All-pass Filter, Fractional Sample Delay Implementation: If it is required to delay a digital signal by a number of discrete sample delays this is easily accomplished using delay elements:



Delaying a signal by 3 samples, using simple delay elements.

Using DSP techniques to delay a signal by a time that is an integer number of sample delays $t_s = 1/f_s$ is therefore relatively straightforward. However delaying by a time that is not an integer number of sampling delays (i.e a fractional delay) is less straightforward.

Another method uses a simple first order all pass filter, to "approximately" implement a fractional sampling delay. Consider the all-pass filter:

$$H(z) = \frac{z^{-1} + a}{1 + az^{-1}} \tag{15}$$

To find the phase response, we first calculate:

$$H(e^{j\omega}) = \frac{e^{-j\omega} + a}{1 + ae^{-j\omega}} = \frac{\cos\omega - j\sin\omega + a}{1 + a\cos\omega - ja\sin\omega} = \frac{(a + \cos\omega) - j\sin\omega}{1 + a\cos\omega - ja\sin\omega} \tag{16}$$

and therefore:

$$\angle H(e^{j\omega}) = \tan^{-1}\left(\frac{-\sin\omega}{a + \cos\omega}\right) + \tan^{-1}\left(\frac{a\sin\omega}{1 + a\cos\omega}\right) \tag{17}$$

For small values of x the approximation $\tan^{-1}x \approx x$, $\cos x \approx 1$ and $\sin x \approx x$ hold. Therefore in Eq. 17, for small values of ω we get:

$$\angle H(e^{j\omega}) \approx \frac{-\omega}{a+1} + \frac{a\omega}{1+a} = -\frac{1-a}{1+a}\,\omega = \delta\omega \tag{18}$$

where $\delta = (1-a)/(1+a)$. Therefore at "small" frequencies the phase response is linear, thus giving a constant group delay of δ. Hence if a signal with a low frequency value f_i, where:

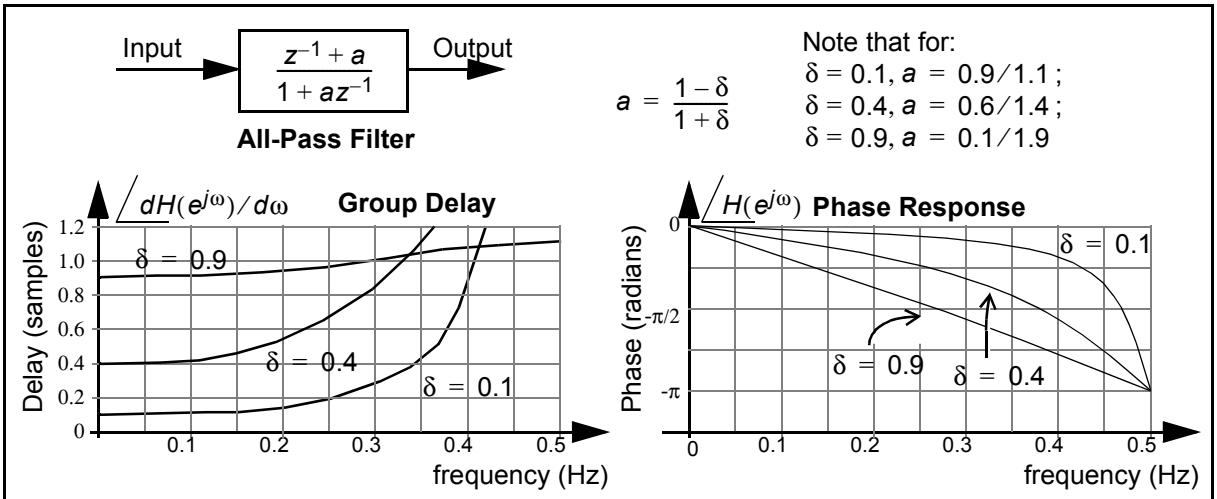
$$\frac{2\pi f_i}{f_s} << 1 \tag{19}$$

is required to be delayed by δ of a sample period ($t_s = 1/f_s$), then:

$$\Rightarrow \delta = \frac{1-a}{1+a} \qquad \Rightarrow a = \frac{1-\delta}{1+\delta} \tag{20}$$

Therefore for the sine wave input signal of $x(k) = \sin(2\pi f_i k / f_s)$ the output signal is approximately $y(k) \approx \sin((2\pi f_i (k-\delta))/f_s)$.

Parameters associated with creating delays of 0.1, 0.4, and 0.9 are shown below :

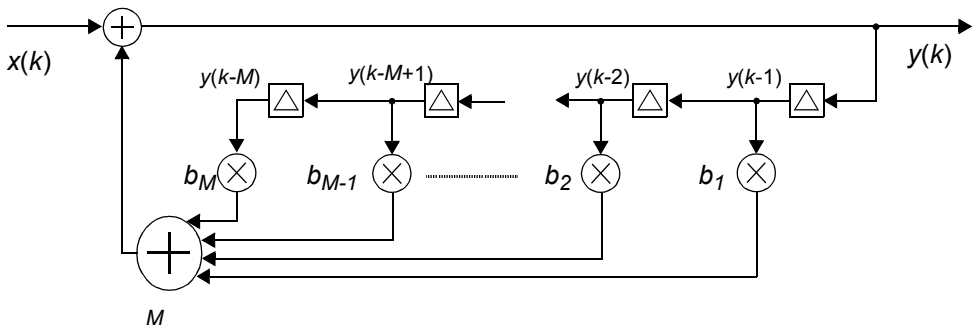


Phase response and group delay for a first order all pass filter implementing a fractional delay at low frequencies. For frequencies below $0.1 f_s$ the phase response is "almost" linear, and therefore the group delay is effectively a constant. Note of course that for a stable filter, $a < 1$. The gain at all frequencies is 1 (a feature of all pass filters of course).

One area where fractional delays are useful is in musical instrument synthesis where accurate control of the feedback loop delay is desirable to allow accurate generation of musical notes with rich harmonics using "simple" filters [43]. If a digital audio system is sampling at $f_s = 48000$ Hz then for frequencies up to around 4000 Hz very accurate control is available over the loop delay thus allowing accurate generation of musical note frequencies. More detail on fractional delay method and applications can be found in [97]. See *All-pass Filter-Phase Compensation, Equalisation, Finite Impulse Reponse Filter - Linear Phase.* .

All-Pole Filter: An *all-pole filter* is another name for a digital infinite impulse response (IIR) filter which features only a recursive (feedback) section, i.e. it has no feedforward (non-recursive) finite

impulse response (FIR) section. The signal flow graph and discrete time equations for an *all-pole filter* are:.



$$y(k) = \sum_{n=1} b_n y(k-n)$$
$$= b_1 y(k-1) + b_2 y(k-2) + \ldots + b_{M-1(2)} y(k-M+1) + b_M y(k-M)$$

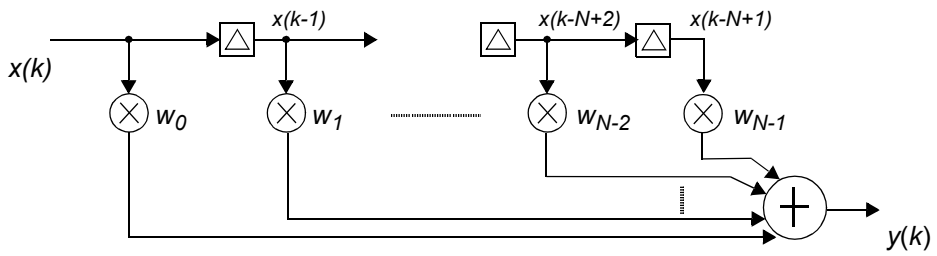
An *all pole filter* has a feedback (recursive) section but ***no*** feedforward (non-recursive) section. As for all IIR filters care must be taken to ensure that the filter is stable and all poles are within the unit circle of the z-domain. (In our example we have used *b*'s to specify the recursive weights, and (where appropriate) *a*'s to specify the non-recursive weights. Some others use precisely the reverse notation!)

An M^{th} order all-pole filter has M weights (b_1 to b_M). and the z-domain transfer function can be represented by an M^{th} order z-polynomial:

$$B(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 + b_1 z^{-1} + \ldots + b_{M-1} z^{-M+1} + b_M z^{-M}}$$
$$= \frac{1}{1 + \sum_{n=1}^{M} b_n z^{-n}} \qquad (21)$$

The all-pole filter weights are also referred to as the autoregressive parameters if the all-pole filter is used to generate an AR process. See also *All-Zero Filter, Autoregressive Model, Autoregressive-Moving Average Filter, Digital Filter, Finite Impulse Response Filter, Infinite Impulse Response Filter*.

All-Zero Filter: An *all zero filter* is another name for a finite impulse response (FIR) digital filter:



$$y(k) = w_0 x(k) + w_1 x(k-1) + w_2 x(k-2) + w_3 x(k-3) + ... + w_{N-1} x(k-N+1)$$

$$= \sum_{n=0}^{N-1} w_n x(k-n) = \boldsymbol{w}^T \boldsymbol{x}_k = \begin{bmatrix} w_0 & w_1 & w_2 \end{bmatrix} \begin{bmatrix} x(k) \\ x(k-1) \\ x(k-2) \end{bmatrix}$$

The signal flow graph and the discrete time output equation for an all zero digital filter. An *all zero filter* is non-recursive and therefore contains no feedback components.

An (N-1)-th order *all-zero filter* has N weights (w_0 to w_{N-1}) and can be represented as an (N-1)-th order polynomial in the z-domain:

$$W(z) = \frac{Y(z)}{X(z)} = w_0 + w_1 z^{-1} + \ldots + w_{N-2} z^{-N+2} + w_{N-1} z^{-N+1} = \sum_{n=0}^{N-1} w_n z^{-n} \qquad (22)$$

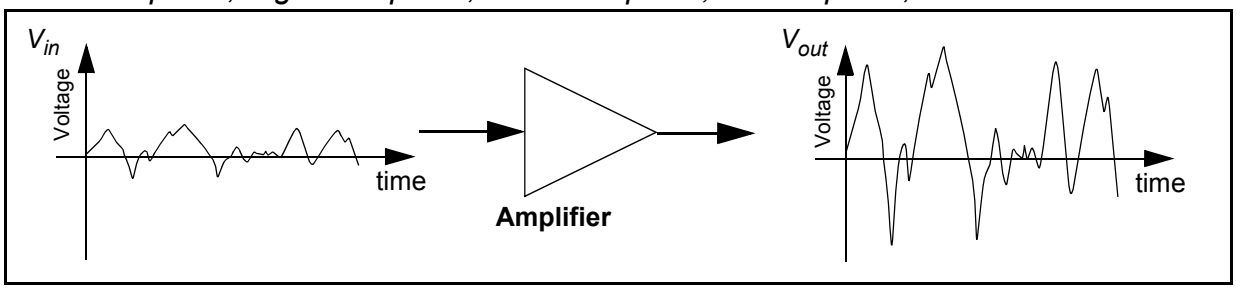
$$= X(z) z^{-N+1} [w_0 z^{N-1} + w_1 z^{N-2} + \ldots w_{N-1}]$$

An *all-zero filter* is often also referred to as a moving average filter, although the name "moving average filter" is (usually) more specifically used to mean an *all-zero filter* where all of the filter weights are 1/N (or 1). See also *All-Pole Filter, Comb Filter, Digital Filter, Finite Impulse Response Filter, Infinite Impulse Response Filter , Moving Average Filter*.

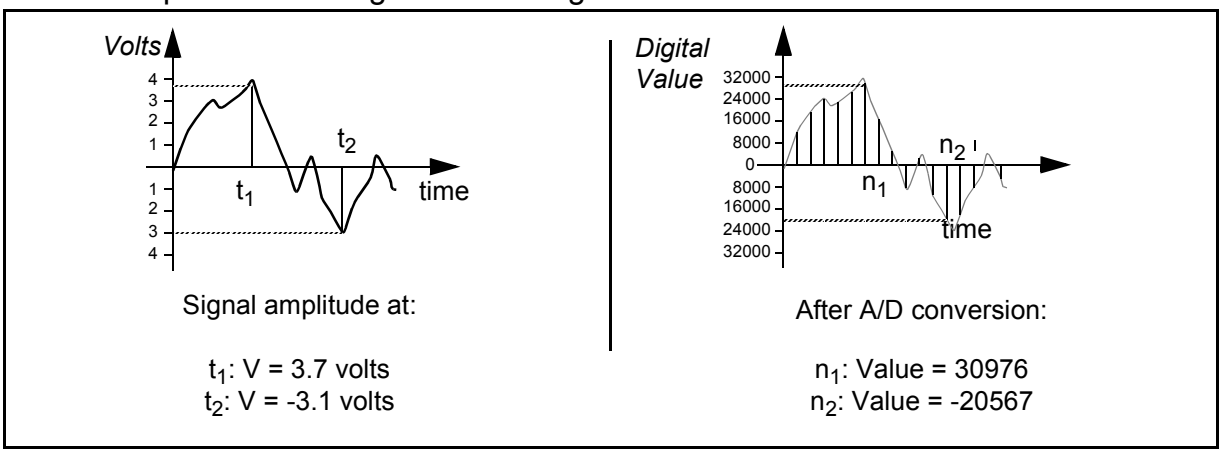
Ambience Processing: The addition of echoes or reverberation to warm a particular sound or mimic the effect of a certain type of hall, or other acoustic environment. Another more popular term used by Hifi companies is Digital Soundfield Processing (DSfP).

Amplifier: A device used to amplify, or linearly increase, the value of an analog voltage signal. *Amplifiers* are usually denoted by a triangle symbol. The amplification *factor* is stated as a ratio V_{out}/V_{in}, or in dBs as $20\log_{10}(V_{out}/V_{in})$. For any real time input/output DSP system some form of amplifier interface is required at the input and the output. A good amplifier should have a very high input impedance, and a very low output impedance. Some systems require an amplification

factor of 1 to protect or isolate a source; this type of *amplifier* is often called a buffer. See also *Operational Amplifier*, *Digital Amplifier*, *Buffer Amplifier, Pre-amplifier*, and *Attenuation*.



Amplitude: The value *size* (or magnitude) of a signal at a specific time. Prior to analog to digital conversion (ADC) the instantaneous *amplitude* will be given as a voltage value, and after the ADC, the *amplitude* of a particular sample will be given as a binary number. Note that a few authors use amplitude as the plus/minus magnitude of a signal.



Amplitude Modulation: One of the three ways of modulating a sine wave signal to carry information. The sine wave or carrier has its amplitude changed in accordance with the information signal to be transmitted. See also *Frequency Modulation*, *Phase Modulation.*

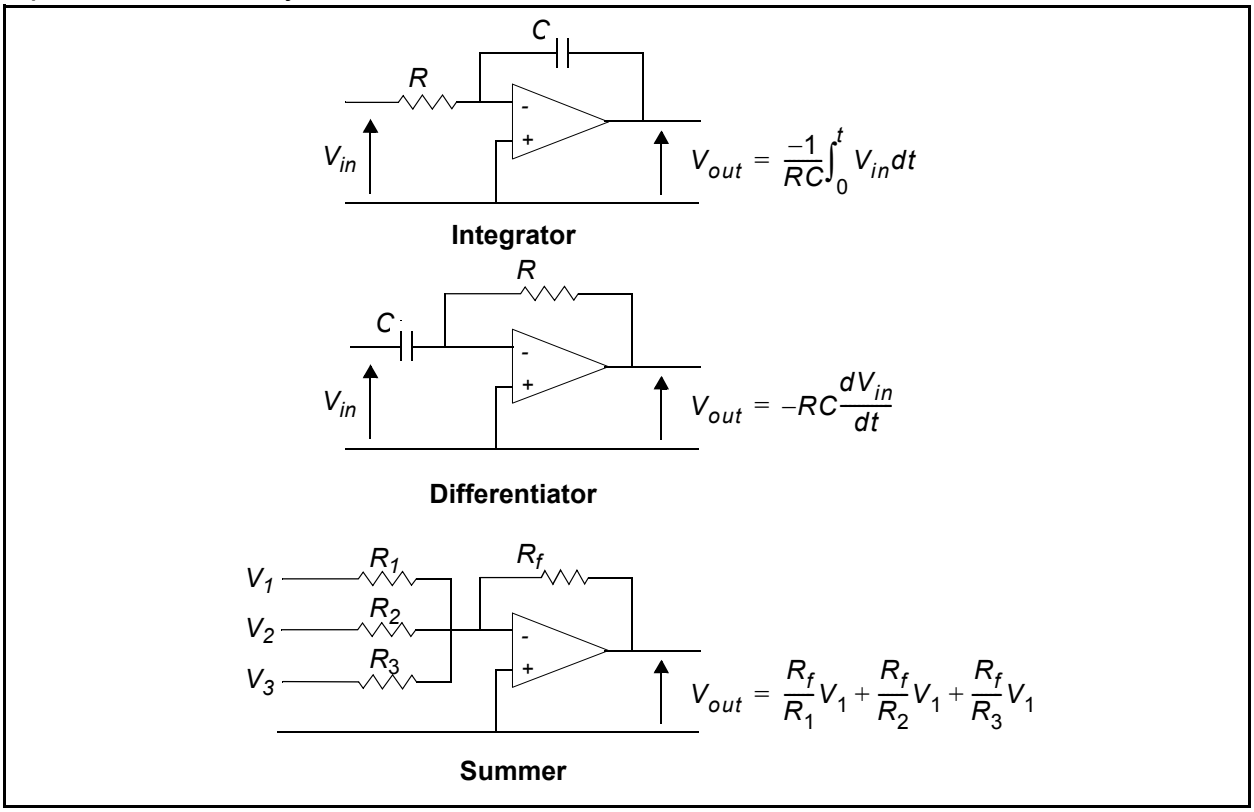
Amplitude Response: See *Fourer Series - Amplitude/Phase Representation, Fourier Series - Complex Exponential Representation*.

Amplitude Shift Keying (ASK): A digital modulation technique in which the information bits are encoded in the amplitude of a symbol. On-Off Keying (OOK) is a special case of *ASK* in which the two possible symbols are zero (Off) and V volts (On). See also *Frequency Shift Keying, Phase Shift Keying, Pulse Amplitude Modulation, Quadrature Amplitude Modulation.*

Analog: An *analog* means the "*same as"*. Therefore, as an example, an *analog* voltage for a sound signal means that the voltage has the same characteristics of amplitude and phase variation as the sound. Using the appropriate sensor, analog voltages can be created for light intensity (a photovoltaic cell), vibrations (accelerometer), sound (microphone), fluid level (potentiometer and floating ball) and so on.

Analog Computer: Before the availability of low cost, high performance DSP processors, *analog computers* were used for analysis of signals and systems. The basic linear elements for *analog computers* were the summing amplifier, the integrator, and the differentiator [44]. By the judicious use of resistor and capacitor values, and the input of appropriate signals, analog computers could

be used for solving differential equations, exponential and sine wave generation and the development of control system transfer functions.

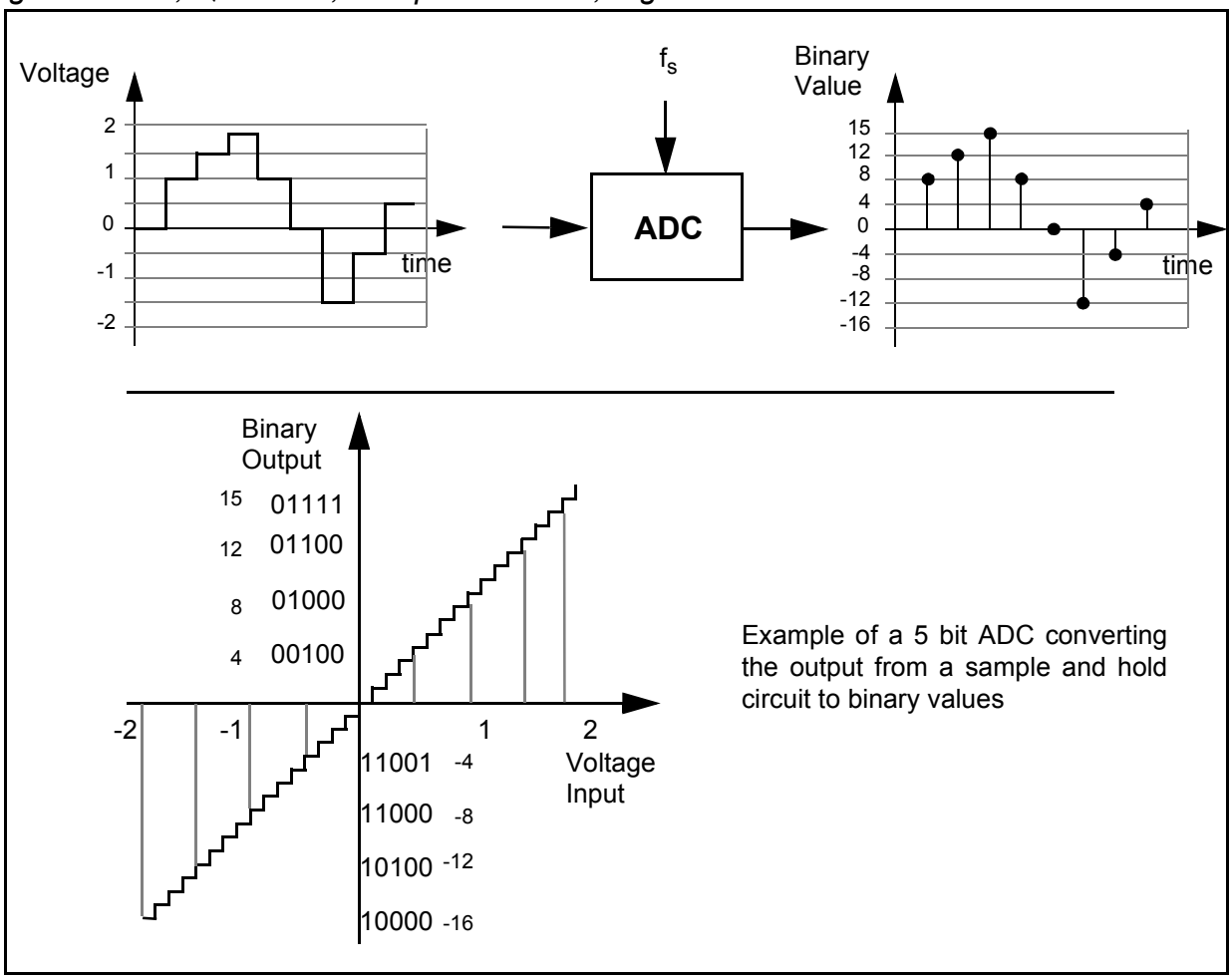


Analog Differentiator: See *Analog Computer.*

Analog Integrator: See *Analog Computer.*

Analog to Digital Converter (A/D or ADC): A *analog to digital converter* takes an analog input voltage (a real number) and converts it (or "quantizes" it) to a binary number (i.e., to one of a finite set of values). The number of conversions per second is governed by the sampling rate. The input to an ADC is usually from a sample and hold circuit which holds an input voltage constant for one sampling period while the ADC performs the actual analog to digital conversion. Most ADCs used in DSP use 2's complement arithmetic. For audio applications 16 bit ADCs are used, whereas for telecommunications and speech coding, 8 bit ADCs are usually used. Modern ADCs can achieve

almost 20 bits of accuracy at sampling rates of up to 100kHz. See also *Anti-alias Filter, Digital to Analog Converter, Quantizer, Sample and Hold, Sigma Delta .*



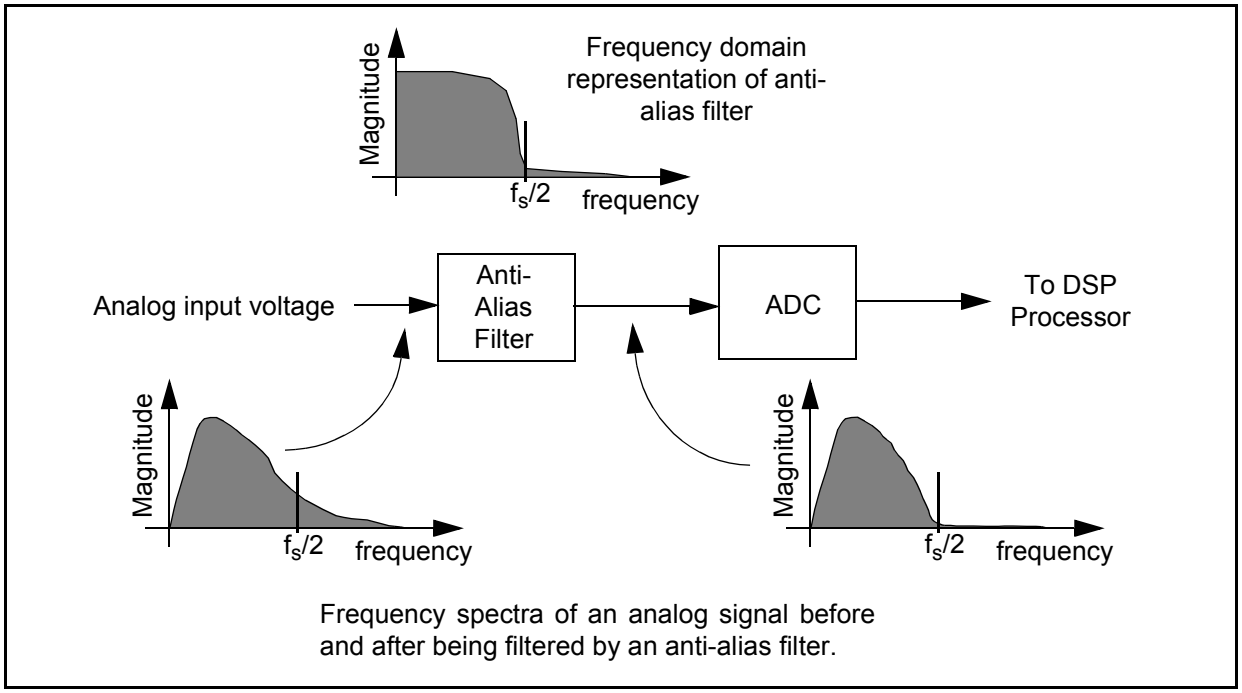
Anechoic: An acoustic condition in which (virtually) no reflected echoes exist. This would occur if two people were having a conversation suspended very high in the air. In practice *anechoic* chambers can be built where the walls are made of specially constructed cones which do not reflect any sound, but absorb it all. Having a conversation in an anechoic chamber can be awkward as the human brain is expecting some echo to occur.

ANSI: American National Standards Institute. A group affiliated with the International Standards Organization (ISO) that prepares and establishes standards for a wide variety of science and engineering applications including transmission codes such as ASCII and companding standards like μ-law, among other things. See also *Standards*.

ANSI/IEEE Standard 754: See *IEEE Standard 754.*

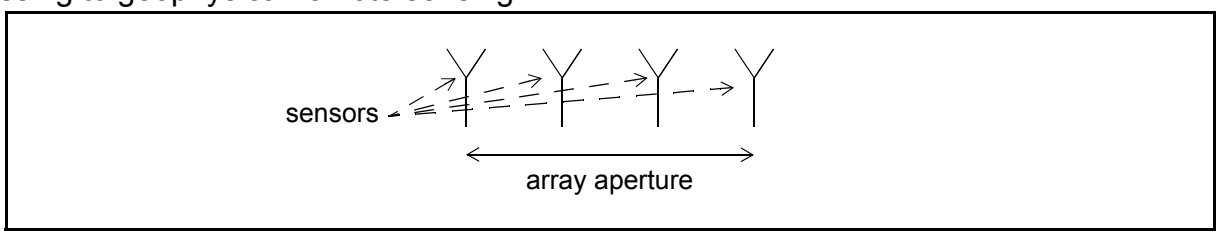
Anti-alias Filter: A filter used at the input to an A/D converter to block any frequencies above $f_s/2$, where f_s is the sampling frequency of the A/D (analog to digital) converter. The *anti-alias* filter is analog and usually composed of resistive and capacitive components to provide good attenuation above $f_s/2$. With the introduction of general *oversampling* techniques and more specifically *sigma-delta* techniques, the specification for analog *anti-alias* filters is traded off against using

oversampling and digital low pass filters. See also *Aliasing, Analog to Digital Converter, Oversampling, Sampling, Sample and Hold*.



Frequency spectra of an analog signal before and after being filtered by an anti-alias filter.

Aperture: The physical distance spanned by an array of sensors or an antenna dish. *Aperture* is a fundamental quantity in DSP applications ranging from RADAR processing to SONAR array processing to geophysical remote sensing.



See also *Beamforming, Shading Weights.*

Aperture Taper: See *Shading Weights.*

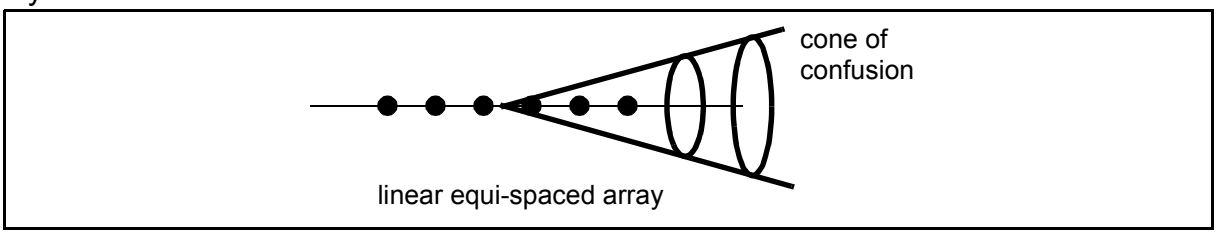
Application Specific Integrated Circuit (ASIC): A custom designed integrated circuit targeted at a specific application. For example, an *ASIC* could be designed that implements a 32 tap digital filter with weights set up to provide high pass filtering for a digital audio system.

Architecture: The hardware set up of a particular DSP system. For example a system which uses four DSP processors, may be referred to as a parallel processing DSP *architecture*. At the chip level, inside most DSP processors a control bus, address bus and data bus are used that is often referred to generically as the Harvard architecture. See also *DSP Board, DSP Processor*.

Arpanet: The name for a US Defense Department's Advanced Research Projects Agency network (circa 1969) which was the first distributed communications network and has now "probably" evolved into the Internet.

Array (1): The name given to a set of quantities stored in a tabular or list type form. For example a 3×5 matrix could be stored as a 3×5 *array* in memory.

Array (2): The general name given to a group of sensors/receivers (antennas, microphones, or hyrophones for example) arranged in a specific pattern in order to improve the reception of a signal impinging on the *array* sensors. The simplest form of *array* is the linear, or 1-D (one dimensional) array which consists of a set of (often equally spaced) sensors. This array can be used to discriminate angles of arrival in any plane containing the array, but is limited because of a cone of confusion. This cone is the cone of angles of arrival that all give rise to identical time differences at the array.



The 2-D *array* has a set of elements distributed in a plane and can be used to discriminate signals in two dimensions of arrival angle. A similar, but less severe confusion results since signals from opposite sides of the plane containing the array (top-bottom) give rise to the same time delays at each of the elements. This may or may not be a problem depending on the geometry of the array and the particular application of the array. 3-D arrays can also be used to eliminate this ambiguity. See also *Beamforming*.

Array Multiplier: See *Parallel Multiplier*.

ASCII: American Standard Code for Information Interchange. A 7 bit binary code that defines 128 standard characters for use in computers and data transmission. See also *EBCDIC*.

Assembler: A program which takes mnemonic codes for operations on a DSP chip, and *assembles* them into machine code which can actually be run on the processor. See also *Cross Compiler, Machine Code.*

Assembly Language: This is a mnemonic code used to program a DSP processor at a relatively low level. The *Assembly language* is then assembled into actual machine code (1's and 0's) that can be downloaded to the DSP system for execution. The *assembly language* for DSP processors from the various DSP chip manufacturers is different. See also *Cross Compiler, Machine Code.*.

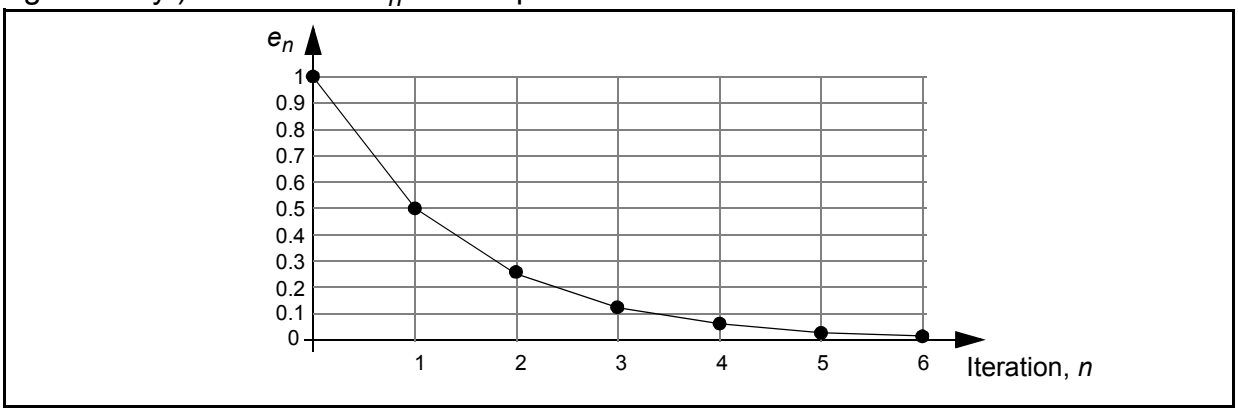
```
movep     y:input, x:(r0)     ; input sample
clr       a                   x:(r0)+,x0    y:(r4)+, y0
rep       #19
mac       x0, y0, a           x:(r0)+,x0    y:(r4)+, y0
macr      x0, x0, a           r0)-
movep     a, y:output         ; output filtered sample
```

A segment of Motorola DSP56000 assembly language to realize a 20 tap FIR filter

Asymptotic: When a variable, x, converges to a solution m, with the error $e = x - m$ reducing with increasing time, but never (in theory) reaching exactly m, then the convergence is *asymptotic*. For example the function:

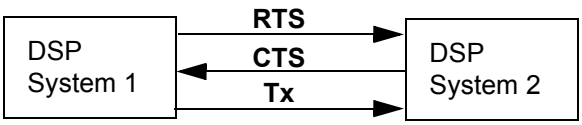
$$x_n = 2^{-n} \tag{23}$$

will *asymptotically* approach zero as *n* increases, but will never reach *exactly* zero. (Of course, if finite precision arithmetic is used then the quantization error may allow this particular result to converge exactly.)The function x_n can be plotted as:



See also *Adaptive Signal Processing, Convergence, Critically Damped, Overdamped, Underdamped*.

Asynchronous: Meaning not synchronized. An *asynchronous* system does not work to the regular beat of a clock, and is likely to use handshaking techniques to communicate with other systems. See also *Handshaking.*

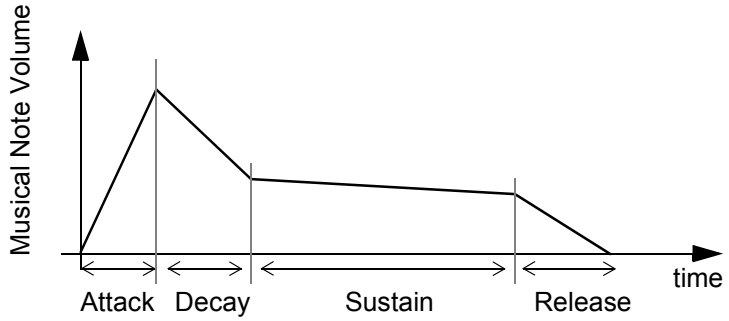


A simple protocol for handshaking. DSP system 1 send an RTS signal (request to send data) to DSP System 2, which replies with a CTS signal (clear to send data) if it is ready to receive data. After the *handshake* using RTS and CTS, the data can be transmitted on the Tx line.

Asynchronous Transfer Mode (ATM): A protocol for digital data transmission (e.g., voice or video data) that breaks data from higher levels in a network into 53 byte cells comprising a 5 byte header and 48 data bytes. The protocol allows for virtual circuit connections (i.e., like a telephone circuit) and can be used to support a datagram network (i.e., like some electronic mail systems). In spite of the word *Asynchronous,* ATM can be used over the ubiquitous synchronous optical network (SONET).

Attack-Decay-Sustain-Release (ADSR): In general the four phases of the sound pressure level envelope of a musical note comprise: (1) the attack, when the note is played; (2) the decay when the note starts to reduce in volume from its peak; (3) the sustain where the note holds its volume and the decay is slow and; (4) the release after the note is released and the volume rapidly decays

away. The *ADSR* profile of most musical instruments is different and varies widely for different classes of instrument such as woodwind, brass, and strings.



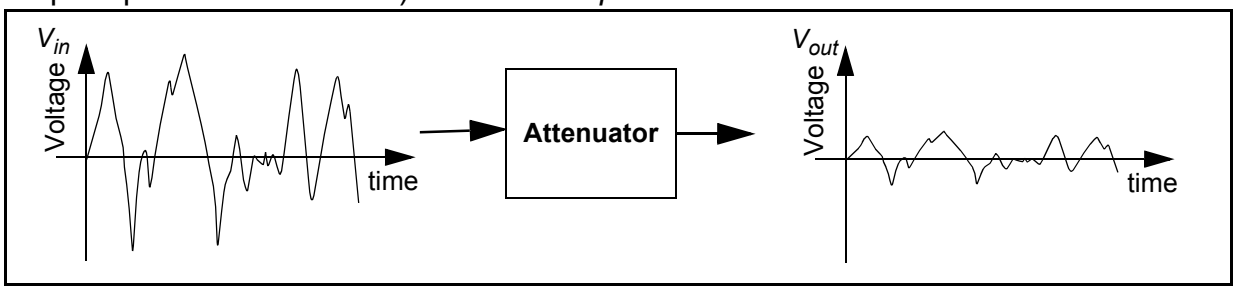
The amplitude envelope of a musical instrument can usually be characterized by four different phases. The relative duration of each phase depends of course on the instrument being played.

Specification of the ADSR values is a key element for synthesizing of musical instruments. See also *Music, Music Synthesis*.

Attenuation: A signal is *attenuated* when its magnitude is reduced. *Attenuation* is often measured as a (modulus) ratio $\left|(V_{out}/V_{in})\right|$, or in dBs as $20\log_{10}\left|V_{out}/V_{in}\right|$. Note that an attenuation of 10 is equivalent to a gain of 10, expressed in dB, an attenuation of 20dB is equivalent to a gain of -20dB, i.e.,

$$\text{Attenuation Factor} = \frac{1}{\text{Gain Factor}} \text{ or Attenuation (dB) = } -\text{Gain (dB)} \qquad (24)$$

Therefore an attenuation factor of 0.1, is actually a gain factor of 10! The simplest form of attenuator for analog circuits is a resistor bridge. Of course, to avoid loading the source it is more advisable to use an op-amp based attenuator.) See also *Amplifier*.



Audio: *Audio* is the Latin word for "I hear" and usually used in the context of electronic systems and devices that produce and affect what we hear.

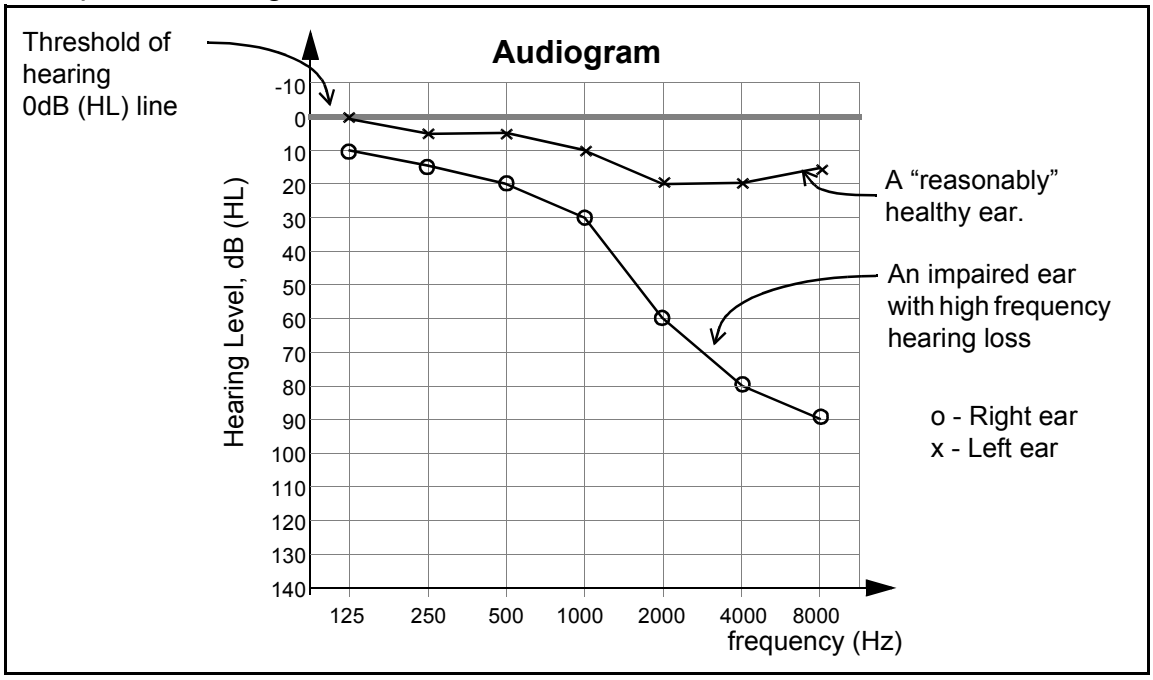
Audio Evoked Potential: See *Evoked Potentials*.

Audio Engineering Society/ European Broadcast Union (AES/EBU): The AES/EBU is the acronym used to describe a popular digital audio standard for bit serial communications protocol for transmitting two channels of digital audio data on a single transmission line. The standard requires the use of 32kHz, 44.1kHz or 48kHz sample rates. See also *Standards*.

Audio Engineering Society (AES): The *Audio Engineering Society* is a professional organization whose area of interest is all aspects of audio. The international headquarters are at 60 East 42nd

Street, Room 2520, New York, NY 10165-2520, USA. The British is at AES British Section, Audio Engineering Society Ltd, PO Box 645, Slough SL1 8BJ, UK.

Audiogram: An *audiogram* is a graph showing the deviation of a person's hearing from the defined "average threshold of hearing" or "Hearing Level". The *audiogram* plots hearing level, dB (HL), against logarithmic frequency for both ears. dB (HL) are used in preference to dB (SPL) - sound pressure level - in order to allow a person's hearing profile to be compared with a straight line average unimpaired hearing threshold.



An *audiogram* is produced by an audiologist using a calibrated audiometer to find the lowest level of aural stimuli just detectable by a patient's left and right ear respectively. See also *Audiometry, Auditory Filters, Ear, Equal Loudness Contours, Frequency Range of Hearing, Hearing Impairment, Hearing Level, Permanent Threshold Shift, Sound Pressure Level, Temporary Threshold Shift, Threshold of Hearing*.

Audiology: The scientific study of hearing. See also *Audiometry, Auditory Filters, Beat Frequencies, Binaural Beats, Binaural Unmasking, Dichotic, Diotic, Ear, Equal Loudness Contours, Equivalent Sound Continuous Level, Frequency Range of Hearing, Habituation, Hearing Aids, Hearing Impairment, Hearing Level, Loudness Recruitment, Psychoacoustics, Sensation Level, Sound Pressure Level, Spectral Masking, Temporal Masking, Temporary Threshold Shift, Threshold of Hearing*.

Audiometer: An instrument used to measure the sensitivity of human hearing using various forms of aural stimuli at calibrated sound pressure levels (SPL). An *audiometer* is usually a desktop instrument with a selection of potentiometric sliders, dials and switch controls to specify the frequency range, signal characteristics and intensity of various aural stimuli. *Audiometers* connect to calibrated headphones (for air conduction tests) or a bone-phone (to stimulate the mastoid bone behind the ear with vibrations if tests are being done to detect the presence of nerve deafness). Occasionally free-field loudspeaker tests may be done using narrowband frequency modulated tones or warble tones. (If pure tones were used nodes and anti-nodes would be set up in the test room at various points).

The most basic form of *audiometer* is likely to only produce pure tones over a frequency range of 125Hz, 250Hz, 500Hz, 1000Hz, 2000Hz, 4000Hz, and 8000Hz. More complex *audiometers* will be able to produce intermediate frequencies and also frequency modulated (FM) or warble tones, bandlimited noise, and spectral masking noise. Because of the dynamic range of human hearing and the severity of some impairments, an *audiometer* may require to be able to generate tones over a 130dB (SPL) range.

Computer based, DSP *audiometers* are likely to completely displace the traditional analogue electronic instruments over the next few years. DSP *audiometers* may even be integrated into PC notebook style "DSP Audiometric Workstations", capable of all forms of audiometric testing, hearing aid testing, and programming of the impending future generation of DSP hearing aids. See also *Audiogram, Audiometry, Auditory Filters, Frequency Range of Hearing, Hearing Impairment, Hearing Level, Sound Pressure Level, Spectral Masking, Threshold of Hearing*.

Audiometry: *Audiometry* is the measurement of the sensitivity of the human ear [30], [157]. For audiometric testing, audiologists use electronic instruments called audiometers to generate various forms of aural stimuli.

A first test of any patient's hearing is usually done with pure tone *audiometry*, using tones with less than 0.05% total harmonic distortion (THD) at test frequencies of 125Hz, 250Hz, 500Hz, 1000Hz, 2000Hz, 4000Hz and 8000Hz and dynamic ranges of almost 130dB (SPL) for the most sensitive human hearing frequencies between 2-4kHz. Each ear is presented with a tone lasting (randomly) between 1 and 3 seconds; the randomness avoids giving rhythmic clues to the patient. The loudness of the tones are varied in steps of 5 and 10dB until a threshold can be determined. The patient indicates whether a tone was heard by clicking a switch. As an example of a test procedure, the British Society of Audiology Test B [157] determines the threshold at a particular frequency as follows:

1. Reduce the tone level in 10dB steps until the patient no longer responds;
2. Three further tones are presented at this level. If none or only one of these is heard, that level is taken as unheard;
3. If all tones in stage 2 were heard, the level is reduced by 5dB until the level is unheard, by repeating stage 2 procedure;
4. If stage 2 was not heard the level is raised by 5dB and as many tones are presented as are necessary to deduce whether at least 2 out of 4 presentations were heard. If this level is heard it is taken as the threshold for that frequency;
5. If stage 4 was not heard the level is raised by 5dB and stage 4 is repeated until a threshold is found;

The results of an audiometric test are usually plotted as an audiogram, a graph of dB Hearing Level (HL) versus logarithmic frequency.

A audiometric procedure using (spectral) masking is particularly important where one ear is suspected to be much more sensitive than the other. Most audiometers will provide a facility to produce spectral masking noise. Masking noise is generally white and is played into the ear that is not being tested in cases where the tone presented to the test ear is very loud. If masking was not used the conduction of the tone through the skull is heard by the other ear giving a false impression about the sensitivity of the ear under test.

More complex audiometers provide a wider range of frequencies, and also facilities for producing narrowband frequency modulated tones, narrowband noise, white noise, and speech noise, thus providing for a more comprehensive facility for investigation of hearing loss. Audiometry is specified

in IEC 645, ISO 6189: 1983, ISO 8253: 1989. See also *Audiogram, Audiology, Ear, Frequency Range of Hearing, Hearing Impairment, Sensation Level, Sound Pressure Level, Spectral Masking, Temporal Masking, Threshold of Hearing*.

Auditory Filters: It is conjectured that a suitable model of the front end of the auditory system is composed of a series of overlapping bandpass filters [30]. When trying to detect a signal of interest in broadband background noise the listener is thought to make use of a filter with a centre frequency close to that of the signal of interest. The perception to the listener is that the background noise is somewhat filtered out and only the components within the background noise that lie in the auditory filter passband remain. The threshold of hearing of the signal of interest is thus determined by the amount of noise passing through the filter.

This *auditory filter* can be demonstrated by presenting a tone in the presence of noise centered around the tone and gradually increasing the noise bandwidth while maintaining a constant noise power spectral density. The threshold of the tone increases at first, however starts to flatten off as the noise increases out with the bandwidth of the auditory filter. The bandwidth at which the tone threshold stopped increasing is known as the critical bandwidth (CB) or equivalent rectangular bandwidth (ERB). These filters are often assumed to have constant percent critical bandwidths (i.e., constant fractional bandwidths). For normal hearing individuals this bandwidth may be about 18 percent -- so an auditory filter centered at 1000 Hz would have a critical bandwidth of about 180 Hz. The entire hearing range can be covered by about 24 (non-overlapping) critical bandwidths. See also *Audiology, Audiometry, Ear, Fractional Bandwidth, Frequency Range of Hearing, Psychoacoustics, Spectral Masking, Temporal Masking, Threshold of Hearing*.

Aural: Relating to the process of hearing. The terms monaural and binaural are related to hearing with one and two ears respectively. See also *Audiology, Binaural, Ear, Monaural, Threshold of Hearing.*

Auralization: The acoustic simulation of virtual spaces. For example simulating the sound of a stadium (an open sound with large echo and long reverberation times) in a small room using DSP.

Autocorrelation: When dealing with stochastic (random) signals, *autocorrelation,* $r(n)$, provides a measure of the randomness of a signal, $x(k)$ and is calculated as:

$$r(n) = E\{x(k)x(k+n)\} = \sum_{k} x(k)x(k+n)p\{x(k), x(k+n)\} \tag{25}$$

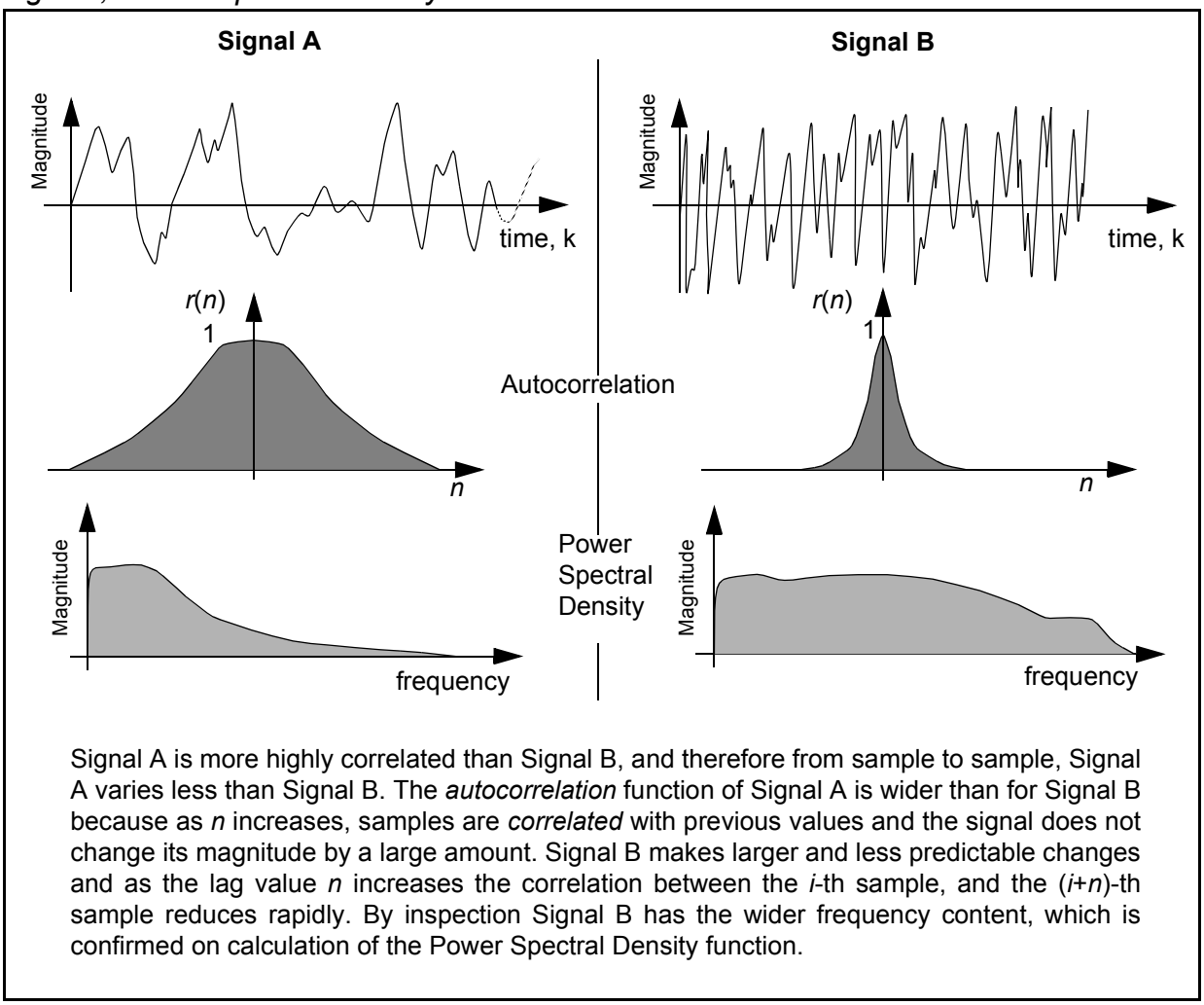
where $p\{x(k), x(k+n)\}$ is the joint probability density function of the signal or random process, $x(k)$ at times *k* and *k*+*n*. For ergodic signals using $2M$ available samples the *autocorrelation* can be estimated as a time average:

$$r(k) = \frac{1}{M} \sum_{k=0}^{M-1} x(n)x(n+k) \quad \text{for large } M \tag{26}$$

If the mean and *autocorrelation* of a signal are constant then the signal is said to be wide sense stationary. In many least mean squares DSP algorithms the assumption of wide sense stationarity is necessary for algorithm derivations and proofs of convergence.

If a signal is highly correlated from sample to sample, then for a particular sample at time, *i*, the next sample at time *i+1* will have a value that can be predicted with a small amount of error. If a signal has almost no sample to sample correlation (almost white noise) then the sample value at time *i+1* cannot be reliably predicted from values of the sequence occurring at or before time *i*. Calculating the *autocorrelation* function, $r(n)$, therefore gives a measure of how well correlated ("or similar") a signal is with itself by comparing the difference between samples at time lags of n = 0,1,2,... and so on.

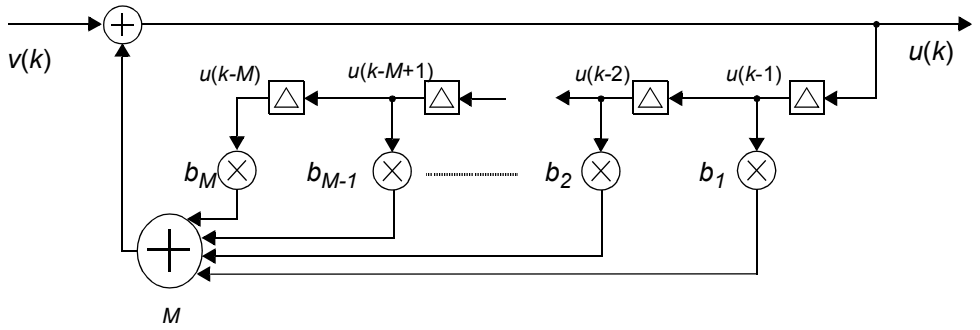
Taking the discrete Fourier transform of the *autocorrelation* function yields the Power Spectral Density (PSD) function which gives a measure of the frequency content of a stochastic signal. See also *Ergodic, Power Spectral Density.*



Signal A is more highly correlated than Signal B, and therefore from sample to sample, Signal A varies less than Signal B. The *autocorrelation* function of Signal A is wider than for Signal B because as *n* increases, samples are *correlated* with previous values and the signal does not change its magnitude by a large amount. Signal B makes larger and less predictable changes and as the lag value *n* increases the correlation between the *i*-th sample, and the (*i*+*n*)-th sample reduces rapidly. By inspection Signal B has the wider frequency content, which is confirmed on calculation of the Power Spectral Density function.

Autoregressive (AR) Model: An *autoregressive model* is a means of generating an *autoregressive* stochastic process. *Autoregressive* refers to the fact that the signal is the output of a all-pole infinite impulse response (IIR) filter that has been driven by white noise input [17], [90].

An autoregressive process can be generated by the signal flow graph and discrete time equations below:



$$u(k) = \sum_{n=1} b_n v(k-n)$$
$$= b_1 v(k-1) + b_2 v(k-2) + \ldots + b_{M-1} v(k-M+1) + b_M v(k-M)$$

An *autoregressive model* has a feedback (recursive) section but no feedforward (non-recursive) section. The input signal, $v(k)$, is assumed to be white Gaussian noise. |The output signal, $u(k)$, is referred to as an autoregressive process. When setting the filter weights values, $\{b_n\}$ care must be taken to ensure that the filter is stable and all filter poles are within the unit circle of the z-domain. In addition, since the autoregressive model is generated with a feedback system, it is necessary to let the AR system reach steady state before using the output samples.

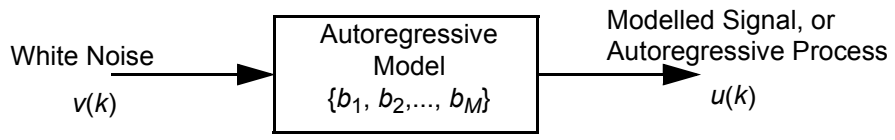
An M^{th} order *autoregressive model* is generated from an all-pole digital filter that has M weights (b_1 to b_M). These weights are also referred to as the autoregressive parameters. The z-domain transfer function can be represented by an M^{th} order z-polynomial:

$$H(z) = \frac{U(z)}{V(z)} = \frac{1}{1 + b_1 z^{-1} + \ldots + b_{M-1} z^{-M+1} + b_M z^{-M}}$$
$$= \frac{1}{1 + \sum_{n=1}^{M} b_n z^{-n}} \qquad (27)$$

If a stochastic signal is produced by using white noise as an input to an all-pole filter, then this is referred to as *autoregressive modelling*. The name “autoregressive” comes from the Greek prefix “auto-” meaning, self or one’s own, and “regression” meaning previous or past, hence the combined meaning of a process whose output is generated from its own past outputs. *Autoregressive models* are sometimes loosely referred to as all-pole models. In addition, sometimes the input to the all-pole model is something other than white noise. For example, in modelling voiced speech a pulse train with the desired pitch period drives the all-pole model.

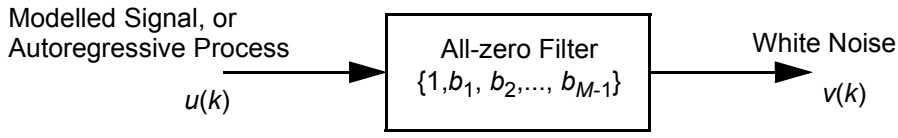
Autoregressive models are widely used in speech processing and other DSP applications whereby a stochastic signal is to be modelled by taking the output of an all-pole filter driven by a stochastic signal. See also *All-Zero Filter, Autoregressive Modelling, Autoregressive-Moving Average Filter, Digital Filter, Infinite Impulse Response Filter*.

Autoregressive Modelling (inverse): Given an *M*-th order autoregressive process the inverse problem is to generate the AR model parameters which can be used to produce this process from a white noise input:

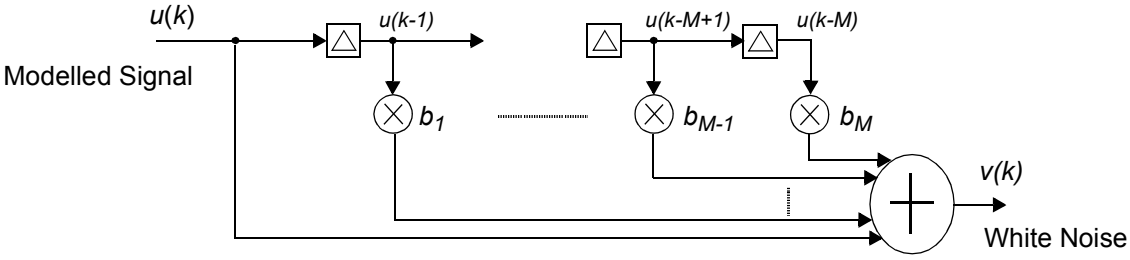


The output signal $u(k)$ is referred to as an autoregressive process, and was generated by a white noise input at $v(k)$. The autoregressive coefficients can be found using statistical signal processing least squares techniques such as Yule-Walker or the LMS algorithm.

To do this, one common approach uses the AR process as the input to an *M*-th order (or greater) all-zero filter with weights {1, b_1, b_2, ... b_M}. If the *M* adjustable weights are selected to minimize the output power, the output will be white noise process. In addition, the feed-forward coefficients from the all-zero model will correspond the parameters of the autoregressive input process. This use of an adaptive FIR predictor is referred to as *autoregressive modelling* [6], [10], [17]:



The white noise signal $v(k)$ can be reproduced by using the modelled stochastic signal as an input to an all zero (FIR) filter with *M* weights, the first weight being 1.



Generation of white noise from an autoregressive process using an all-zero filter.

To see that the AR parameters are recovered we can rewrite Eq. 27 (see *Autoregressive Model*) as:

$$V(z) = \frac{U(z)}{H(z)} = 1 + b_1 z^{-1} + \ldots + b_{M-1} z^{-M+1} + b_M z^{-M} \tag{28}$$

If a given stochastic signal, $u(k)$ was in fact generated by an autoregressive process then we can use mean square minimization techniques to find the autoregressive parameters (i.e., the all-pole filter weights) that would produce that signal from a white noise input. First note that the output of the all zero filter is given by:

$$v(k) = u(k) + \sum_{m=1}^{M} b_m u(k-m) = u(k) + \boldsymbol{b}^T \boldsymbol{u}(k-1) \tag{29}$$

where the vector $\boldsymbol{b} = [b_1 \ \ldots \ b_{M-1} \ \ b_M]^T$

and the vector $\boldsymbol{u}(k) = [u(k-1) \ \ldots \ u(k-M+1) \ \ u(k-M)]^T$

If we attempt to minimize the signal $v(k)$ at the output of the filter, then this is implicitly done by generating the predictable components present in the stationary stochastic signal $u(k)$ (assuming the filter is of sufficient order) which means that the output $v(k)$ will consist of the completely unpredictable part of the signal which is, in fact, white noise (See *Wold Decomposition* and [17]).

To use MMSE techniques, first note that the squared output signal is:

$$\begin{aligned} v^2(k) &= [u(k) + \boldsymbol{b}^T \boldsymbol{u}(k-1)]^2 \\ &= u^2(k) + [\boldsymbol{b}^T \boldsymbol{u}(k-1)]^2 + 2u(k)\boldsymbol{b}^T \boldsymbol{u}(k-1) \\ &= u^2(k) + \boldsymbol{b}^T \boldsymbol{u}(k-1)\boldsymbol{u}^T(k-1)\boldsymbol{b} + 2\boldsymbol{b}^T[u(k)\boldsymbol{u}(k-1)] \end{aligned} \tag{30}$$

Taking expected (or mean) values using the expectation operator $E\{.\}$ we can write the mean squared value, $E\{v^2(k)\}$ as:

$$E\{v^2(k)\} = E\{u^2(k)\} + \boldsymbol{b}^T E\{\boldsymbol{u}(k-1)\boldsymbol{u}^T(k-1)\}\boldsymbol{b} + 2\boldsymbol{b}^T E\{u(k)\boldsymbol{u}(k-1)\} \tag{31}$$

Writing in terms of the $M \times M$ correlation matrix,

$$\boldsymbol{R} = E\{\boldsymbol{u}(k-1)\boldsymbol{u}^T(k-1)\} = \begin{bmatrix} r_0 & \cdots & r_{M-2} & r_{M-1} \\ \vdots & \cdots & \vdots & \vdots \\ r_{M-2} & \cdots & r_0 & r_1 \\ r_{M-1} & \cdots & r_1 & r_0 \end{bmatrix} \tag{32}$$

and the $M \times 1$ correlation vector,

$$\boldsymbol{r} = E\{u(k)\boldsymbol{u}(k-1)\} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_M \end{bmatrix} \tag{33}$$

where $r_n = E\{u(k)u(k-n)\} = E\{u(k-n)u(k)\}$.

gives,

$$E\{v^2(k)\} = E\{u^2(k)\} + \boldsymbol{b}^T \boldsymbol{R}\boldsymbol{b} + 2\boldsymbol{b}^T \boldsymbol{r} \tag{34}$$

Given that this equation is quadratic in $\boldsymbol{b}$ then there is only one minimum value (See entry for *Wiener-Hopf Equations* for more details on quadratic surfaces). The minimum mean squared error (MMSE) solution occurs when the predictable component in the signal $u(k)$ is completely predicted, leaving only the unpredictable white noise as the output. This yields the autoregressive components, $\boldsymbol{b}_{AR}$, can be found by setting the (partial derivative) gradient vector, ∇, to zero:

$$\nabla = \frac{\partial}{\partial \boldsymbol{b}} E\{x^2(k)\} = 2\boldsymbol{R}\boldsymbol{b}_{AR} + 2\boldsymbol{r} = \boldsymbol{0} \tag{35}$$

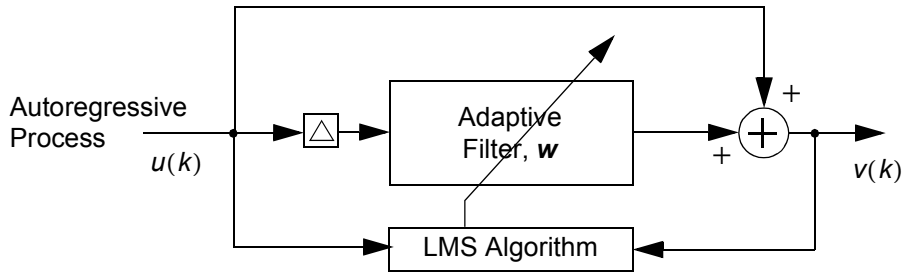
$$\Rightarrow \boldsymbol{b}_{AR} = -\boldsymbol{R}^{-1}\boldsymbol{r} \tag{36}$$

Therefore, given a signal that was generated by an autoregressive process, Eq. 36 (known as the Yule Walker equations) can be used to find the parameters of the autoregressive process, that would generate the signal $u(k)$ given a white noise input signal, $v(k)$.

To practically calculate Yule Walker equations requires that the $\boldsymbol{R}$ matrix and $\boldsymbol{r}$ vector are realized from the stochastic signal $u(k)$, and the $\boldsymbol{R}$ matrix is then inverted prior to premultiplying vector $\boldsymbol{r}$. Assuming that the signal $u(k)$ is ergodic, then in the real world we can calculate elements of $\boldsymbol{R}$ and $\boldsymbol{r}$ from:

$$r_n \cong \frac{1}{N} \sum_{n=0}^{N-1} u(k)u(k-n) \tag{37}$$

where N is a large number of samples that adequately represent the signal. Clearly, solving the Yule-Walker equations requires a very large number of computations, and is usually not done directly in real time systems (See entry *Wiener-Hopf* for more details). Instead the Levinson-Durbin algorithm is used which is an efficient technique for solving equations of the form of Eq. 36. In many systems the LMS (least mean squares) algorithm [53] is used in a predictor architecture:



$$y(k) = \text{Filter}\{u(k), \boldsymbol{w}(k)\}$$

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu v(k)x(k-1)$$

The signal that was generated by an autoregressive process is input to the delay and thereafter adaptive filter. The adaptive filter attempts to minimize the signal $v(k)$ and will therefore set the coefficients to values such that the periodic component of the signal is predicted by the autoregressive filter weights.

Autoregressive modelling is widely used in speech processing and whereby speech is assumed to be generated by an autoregressive process and by extracting the autoregressive filter weights

(parameters) these can be used for later generation of unvoiced speech components (speech synthesis) or for speech vocoding [11]. For model based speech coding the linear prediction problem of Eq. 36 is solved using the Levinson-Durbin algorithm. For speech coding techniques based on waveform coding, the predictor is more likely to the of the simple LMS form.

Other stochastic linear filter models include the moving average (MA) model and the autoregressive moving average (ARMA) models. However the autoregressive filter is by far the most popular for modelling for the main reasons that to find weights requires the solution of a set of linear equations and that it is a generally good model for many applications. The MA or ARMA models, on the other hand, require the solution of a (more difficult to solve) set of non-linear equations.

See also *Adaptive Filtering, Autoregressive Model, Autoregressive Moving Average Filter, Autoregressive Parametric Spectrum Estimation, Least Mean Squares Algorithm, Moving Average Model.*

Autoregressive Moving Average (ARMA) Model: An *autoregressive moving average model* uses a combination of an autoregressive model and moving average model. If white noise is input to an ARMA model, the output is the desired process signal $u(k)$. Unfortunately solving the equations for an ARMA model requires the solution of a set of non-linear equations. See also *Autoregressive Model*, *Moving Average FIR Filter.*

Autoregressive Parametric Spectral Analysis: Using an autoregressive model we can perform parametric power spectral analysis. From the coefficients of the all-pole filter, we can generate the power spectrum of the autoregressive process output, $u(k)$ (see above figure in *Autoregressive Model*) by exploiting the fact that the white noise input has a flat spectrum and a total power of σ^2 [17], [90].Noting that the filter frequency response is:

$$\begin{aligned} H(f) &= \frac{1}{1 + b_1 e^{-j\omega} + \ldots + b_{M-1} e^{-j|(M-1)\omega} + b_M e^{-jM\omega}} \\ &= \frac{1}{1 + \sum_{n=1}^{M} b_n e^{-j\omega n}} \end{aligned} \tag{38}$$

then the power spectrum of the autoregressive filter output is:

$$|Y(f)|^2 = \sigma^2 |H(f)|^2 \tag{39}$$

(assuming frequency is normalized so f_s=1). See also *Autoregressive Model, Autoregressive Modelling*.

Autoregressive (AR) Power Spectrum: See *Autoregressive Model*.

Autoregressive (AR) Process: See *Autoregressive Model*.

Averaging: See *Waveform Averaging, Exponential Averaging, Moving Average, Weighted Moving Average.*

AZTEC Algorithm: *Amplitude Zone Time Epoch Coding* (*AZTEC*) is an algorithm used for data compression of ECGs. The algorithm very simply decomposes a signal into plateaus and slopes

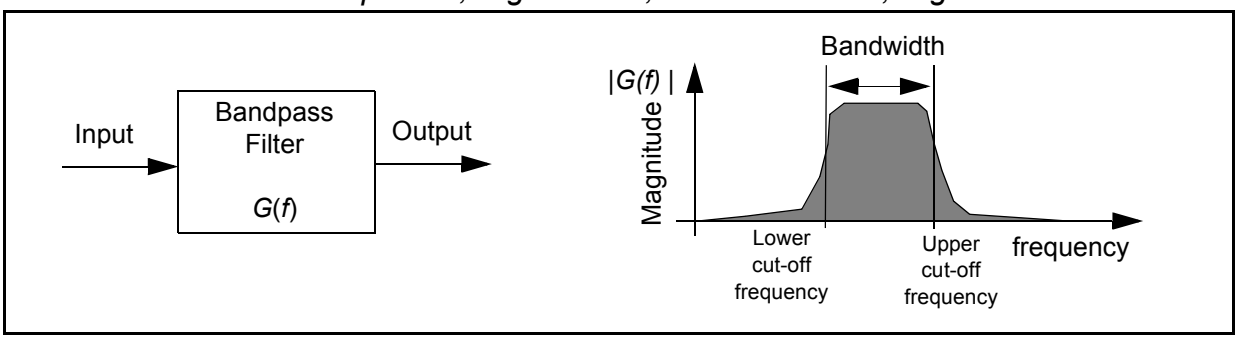
which are then coded in an a data array. Compression ratios of a factor of 10 can be achieved, however the algorithm can cause PRD (Percent Root-mean-square Difference) error levels of almost 30% [48].

B

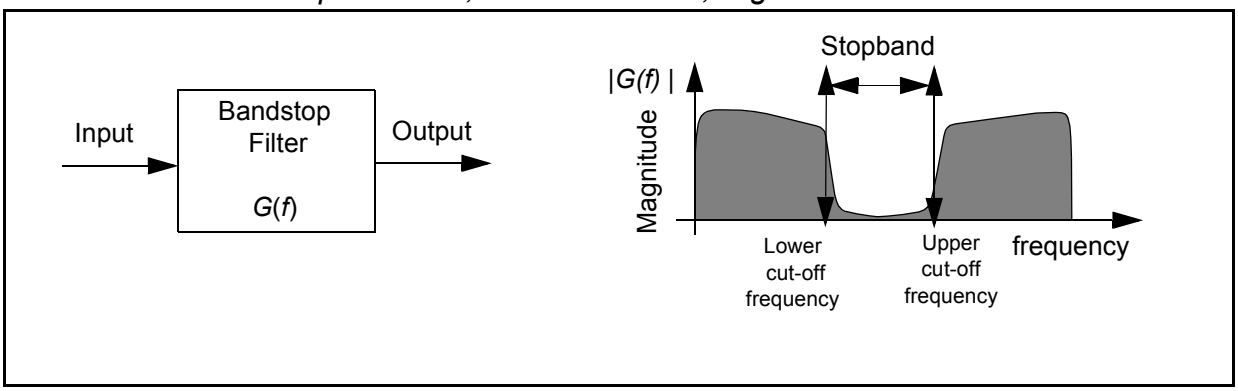
Back Substitution: See *Matrix Algorithms - Back Substitution*.

Band Matrix: See *Matrix Structured - Band.*

Bandpass Filter: A filter (analog or digital) that preserves portions of an input signal between two frequencies. See also *Bandstop Filter, Digital Filter, Low Pass Filter, High Pass Filter*.



Bandstop filter: A filter (analog or digital) that removes portions of an input signal between two frequencies. See also *Bandpass Filter, Low Pass Filter, High Pass Filter.*



Bandwagon: The general English definition is a party, cause or group that people may jump on, or become involved with when it looks likely to succeed. The term was used by the famous information theorist Claude Shannon in 1956 [130] to describe the explosion of interest in his then recently published (1948) information theory paper. In referring to that particular bandwagon Shannon commented that:

> *"Research rather than exposition is the keynote, and our critical thresholds should be raised. Authors should submit only their best efforts, and these only after careful criticism by themselves and their colleagues. A few first rate papers are preferable to a large number that are poorly conceived or half finished. The latter are no credit to their writers and a waste of time to their readers."*

Bartlett Window: See *Window.*

Baseband: Typically, a signal prior to any form of digital or analog modulation. A baseband signal extends from 0Hz contiguously over an increasing frequency range. For example if a radio station produces a *baseband* audio signal (typically music, 0 - 12kHz) in either a digital or analog form, the baseband signal is then modulated onto a carrier (such as 102.5MHz for an FM radio station) for transmission and subsequent reception by radio receivers. At the radio receiver the signal will be

demodulated back to its original frequency band. *Baseband* can also refer to a naturally bandpass signal that has been mixed down to DC.

Basis: See *Vector Properties and Definitions - Basis.*

Basis Function: A periodic signal, $x(t)$,with period T can be expressed as a series of periodic basis functions, $\{\phi_k\}$ such that:

$$x(t) = \sum_{n=-\infty}^{\infty} c_n \phi_n(t) \tag{40}$$

A basis is said to be orthogonal if:

$$\langle \phi_i(t), \phi_i(t) \rangle = \int_a^b \phi_i(\tau)\phi_j^*(\tau)d\tau \quad \text{for } i \neq j \tag{41}$$

where "*" denotes complex conjugate. It is useful to find an orthogonal basis, as if other functions are to be used to approximate a given signal, then it is useful to have as little similarity as possible between the various functions to avoid providing redundant information. The complex exponential used in the Fourier series are an orthogonal set of functions and if $\phi_k(t) = e^{jk\omega_0 t}$ where $\omega_0 = 2\pi/T$ then this is the complex or exponential Fourier series See also Fourier Transform, Matrix Operations.

Baud: A measure of data transmission rate, mean symbols per second. *Baud* is often mis-used to mean bits per second. A *baud* is actually equal to the number of discrete events or transitions per second. There is potential confusion over the proper use of the word *baud* since at high data transmission speeds where data compression techniques are used (V42bis) the number of character bits per second transmitted does not necessarily equal the transmitted data rate in symbols per second.

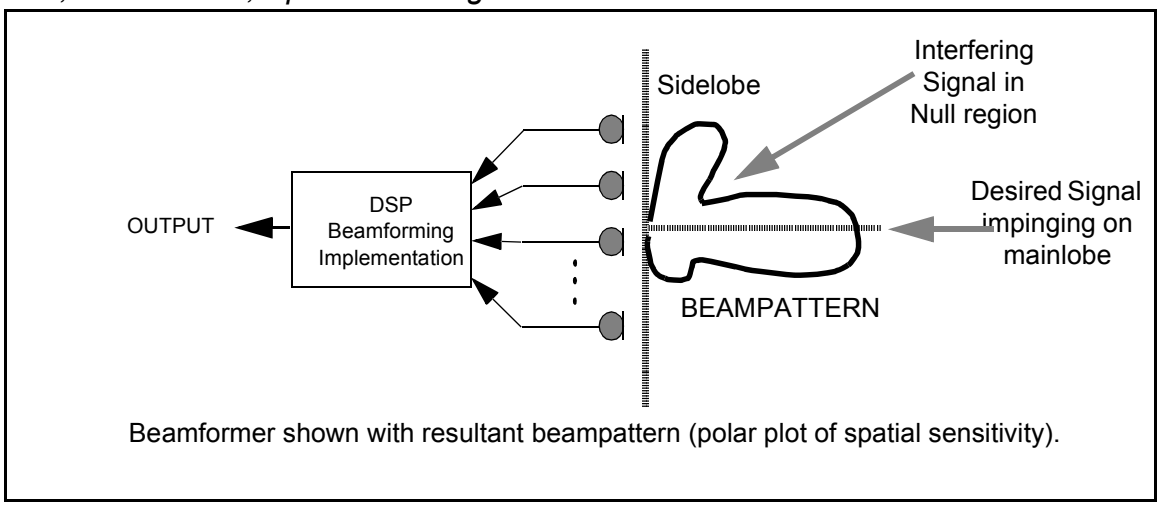
Baugh-Wooley Multiplier: A type of parallel multiplier which operates on 2's complement data and is widely used in DSP [106]. See also *Parallel Multiplier*.

Bayes Theorem: See *Probability*.

Beamforming: A technique to enhance the sensitivity of a device towards a given direction (the look direction) by exploiting the spatial separation of an array of sensors (microphones or hydrophones for example). The array could be a linear 1-D array, 2-D array or even 3-D. The primary motivation behind *beamforming* is often a desire to copy a signal of interest while suppressing spatially disparate interfering signals. Delay-and-sum beamformers simply combine the outputs of a number of sensors (after signals are delayed to allow constructive interference in the look direction).

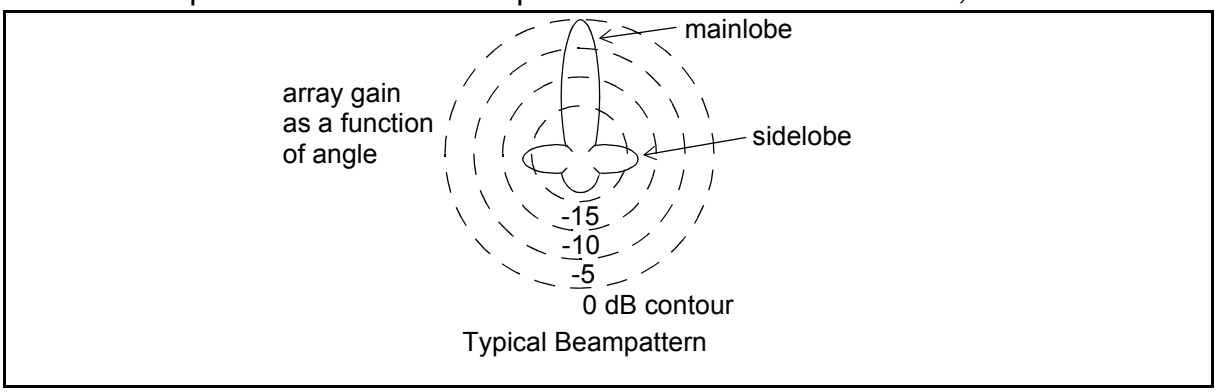
More advanced adaptive *beamforming* techniques go further by attempting to null out any signals arriving from at the array that are not in the desired look direction. The key mechanisms responsible for the spatial sensitivity of a beamformer are constructive and destructive interference. Bearing estimation is related to *beamforming*, but not necessarily the same. A bearing estimator enhances Direction of Arrival (DOA) information for signals of interest, while a beamformer produces an enhanced copy of a signal of interest. See also *Adaptive Beamformer, Bearing Estimation,*

Broadside, Endfire, Constructive Interference, Delay-and-Sum Beamformer, Destructive Interference, Localization, Spatial Filtering.



Beamformer shown with resultant beampattern (polar plot of spatial sensitivity).

Beampattern: A plot of spatial sensitivity of a beamformer (or antenna) as a function of direction. The main lobe and sidelobes are often easily distinguished. Any nulls (direction with virtually no sensitivity) are also clearly distinguished. *Beampatterns* can be plotted for a single frequency (useful for a narrowband application) or as a broadband measure where the sensitivity in each direction is integrated over the frequency span of interest. Broadband patterns seldom contain the deep nulls that are present in narrowband patterns. See also *Beamformer, Localization.*



Typical Beampattern

Bearing Estimation: A classic signal processing problem where it is required to find the angular direction of a number of incoming source signals. In *bearing estimation*, source signal copy is not a concern. See also *Beamforming, Localization.*

Beat Frequencies: When two audible tones of similar frequencies are played together they will effectively go in and out of phase with each other and alternately constructively and destructively interfere. Depending on the frequencies and the magnitude of the difference between the tones they may be aurally perceived as *beat frequencies* rather than two distinct tones. If the frequency difference is no greater than about 10Hz then the ear will follow the amplitude fluctuations and therefore perceive a low beat frequency. *Beat frequencies* are heard most clearly for tones between around 300Hz and 600Hz. As the frequency of the tones increases above 1000-1500Hz the tones will be heard distinctly rather than as beats. This phenomenon is consistent with the fact the neural firings of the auditory system lose synchrony with the incoming sine wave at these frequencies.

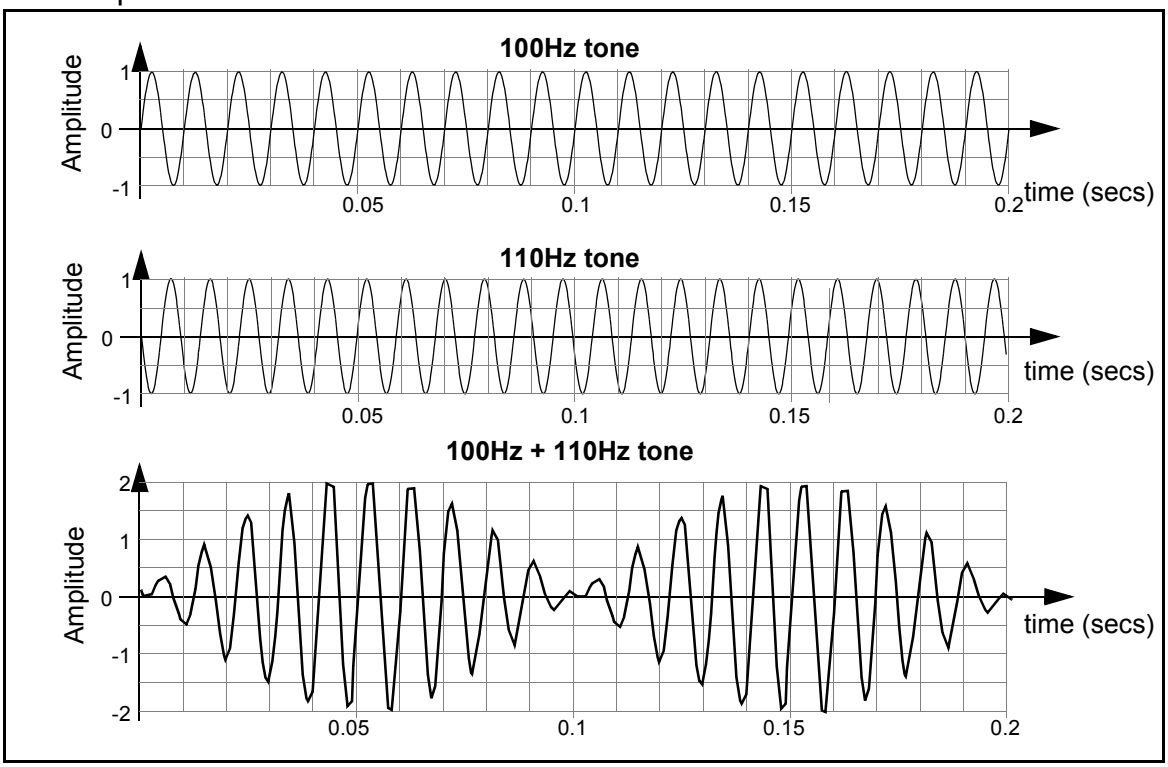
Simple trigonometry shows that:

$$\cos A + \cos B = 2\cos\frac{(A-B)}{2}\cos\frac{(A+B)}{2} \tag{42}$$

Therefore if a 100Hz tone and a 110Hz tone are played simultaneously the composite tone can be written as:

$$\begin{aligned}\cos(2\pi 100t) + \cos(2\pi 110t) &= 2\cos\frac{(2\pi 10t)}{2}\cos\frac{(2\pi 210t)}{2} \\ &= 2\cos(2\pi 5t)\cos(2\pi 105t)\end{aligned} \tag{43}$$

which can be represented as:



The composite tone clearly shows the amplitude fluctuation at 10 times per second caused by the 5Hz modulation effect.

A phenomenon called binaural beats (as distinct from the above description of monaural beats) occurs when a tone of one frequency is presented to one ear, and a slightly different tone frequency is presented to the other ear [30]. The sound will appear to fluctuate at a rate corresponding to the difference between the frequencies. See also *Audiology, Binaural Beats, Binaural Unmasking, Psychoacoustics*.

Bell 103/ 113: The *Bell 103/113* is a modem standard for communication at 300 bits/sec. The *Bell 103/113* is a full duplex modem using FSK (frequency shift keying) modulation. The frequencies used are:

		Originate End (Hz)	Answer End (Hz)
Transmit:	Space	1070	2025
	Mark	1270	2225
Receive:	Space	2025	1070
	Mark	2225	1270

The transmit level is 0 to -12 dBm and the receive level is 0 to -50 dBm.

Although in the mid 1990s modem speeds of 14400 bits/sec are standard and (compressed) bit data rates of 115200 bits/sec are achievable for remote computer communication, the 300 baud modem is still one of the top selling modems! This is due to low rate modems being used for short time connection applications where only a few bytes of data are exchanged, such as telephone credit card verification, traffic light control, remote metering and security systems. See also *Bell 202, Standards, V-Series Recommendations*.

Bell 202: The *Bell 202* is a modem standard for communication at 1200 bits/sec. The *Bell 202* is a half duplex modem using FSK (frequency shift keying) modulation. The frequencies used are:

	Transmit (Hz)
Space	2200
Mark	1200

See also *Bell 103/113, Bell 212, Standards, V-Series Recommendations*.

Bell 212: The *Bell 212* is a modem standard for communication at 1200 bits/sec. The *Bell 202* is a full duplex modem using QPSK (quadrature phase shift keying) modulation. The carrier frequencies used are:

	Originate End (Hz)	Answer End (Hz)
Transmit:	1200	2400
Receive:	2400	1200

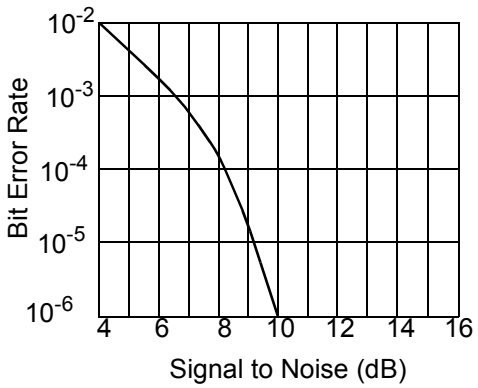
Each keying carries two bits:

Message (2 bits)	Phase Angle
00	90^o
01	0^o
10	180^o
11	270^o

See also *Bell 103/113, Bell 202, Standards, V-Series Recommendations*.

Bento: *Bento* is a multimedia data storage and interchange format the development of which was sponsored primarily by Apple Inc probably with the intention that it would become a de facto standard. The standard is available from `ftp://ftp.apple.com/apple/standards/`. See also *Standards*.

BER vs. S/N Test: (Bit Error Rate vs. Signal to Noise Ratio). A test used to measure the ability of a modem (or a digital communication system) to operate over noise lines with a minimum of data transfer errors. Since even on the best of telephone lines there is always some level of noise, the modem should work with the lowest S/N ratio possible.



Plot of BER vs. S/N for a *typical* modem operating at 1200 bits/second

Other modem performance characteristics include **BER vs. Phase Jitter** which demonstrates the tolerance to phase jitter; **BER vs. Receive Level** which measures the sensitivity to the received signal dynamic range (typically 36dB is the minimum desirable); **BER vs. Carrier Offset** which indicates how the modem performance is affected by the shifts in the carrier frequency encountered in normal public telephone networks (ITU-T specifications allow up to as a 7Hz offset).

Bessel Filter: See *Filters*.

Bidiagonal Matrix: See *Matrix Structured - Bidiagonal.*

Binary: Base 2, where only the digits 0 and 1 are used to represent numbers, e.g.

MSB							LSB	
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
0	1	0	1	1	0	1	0	= 90
1	0	0	0	0	0	0	0	= 128
1	1	0	0	0	0	0	0	= 192

The decimal equivalents of the unsigned 8 bit numbers 01011010, 10000000, and 11000000.

See also *Binary Point, Two's Complement*.

Binary Phase Shift Keying (BPSK): A special case of PSK in which two signals with differing phase exist in the signal set. See also *Phase Shift Keying*.

Binary Point: The *binary point* is the base 2 equivalent of the decimal point. Bits after the *binary point* have a fractional value. See also *Fractional Binary*, *Integer Arithmetic, Two's Complement*. .

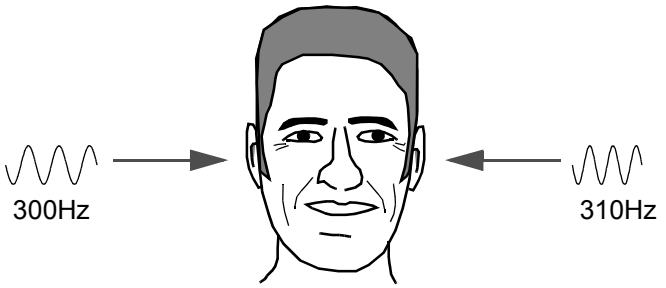
MSB							LSB	
-2^0 •	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	
−1	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.0078125	
0	1	0	1	1	0	1	0	= 0.84375
1	0	0	0	0	0	0	0	= −1
1	1	0	0	0	0	0	0	= −0.5

The decimal equivalents of 0.1011010, 1.0000000, and 1.1000000. Note that the 2's complement notation can still be used, with the most significant bit having a weighting of −1.

Binaural: *Binaural* processing refers to an audio system that processes signals for presentation to two ears. See also *Monaural, Monophonic, Stereophonic.*

Binaural Beats: A phenomenon called *binaural beats* occurs when a tone of one frequency is presented to one ear, and a slightly different tone frequency is presented to the other ear using headphones. The sound will appear to fluctuate at a rate corresponding to the difference between the frequencies. *Binaural beats* are a result of the interaction of the nervous system of the output of the ear to the brain. *Binaural beats* would appear to indicate that the auditory nerve preserves

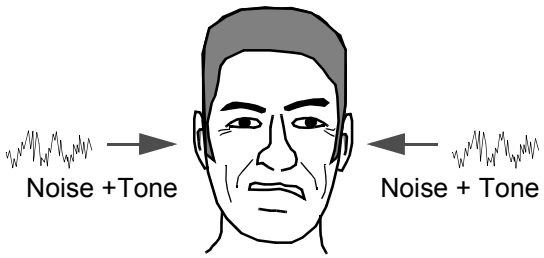
phase information about the acoustic stimulus [30]. See also *Audiology, Beat Frequencies, Binaural Unmasking, Psychoacoustics*.



Listener will experience 10 binaural beats per second.

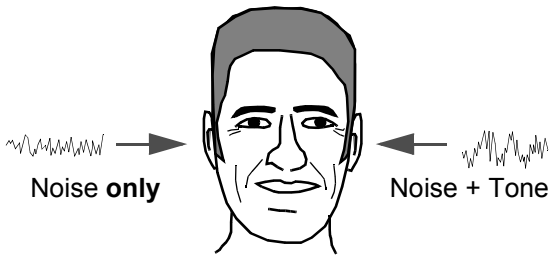
Binaural Unmasking: If a tone masked by white noise is played into one ear or both ears (diotic stimulus) then the auditory mechanism will not perceive the tone without either increasing the tone sound pressure level (SPL) or decreasing the white noise SPL. However if the tone + white noise is played into one ear, and the white noise only into the other ear (dichotic stimulus) then the auditory effect of *binaural unmasking* will actually make the tone more readily detectable.

Binaural unmasking will also occur when noise + tone is input to both ears, but the phase of one the tones is shifted by 180^{o} relative to the other one.)



Tone NOT perceived

The tone in the both ears is completely masked by the white noise and therefore not perceived.



Tone perceived

If noise only is played into the right ear the tone becomes readily detectable. Hence the auditory mechanism is providing a form of noise cancellation.

As a crude DSP analogy, compare this effect to the adaptive noise canceller whereby if a (correlated) noise reference is available the noise in a speech + noise signal can be attenuated, thus providing the improved SNR at the canceller output. See also *Adaptive Noise Cancellation, Audiometry, Dichotic, Diotic.*

Biomedical Signals: Over the last few years *biomedical signals* such as ECGs, EEGs, Evoked Potentials, EMGs have been recorded using DSP acquisition hardware, sampling at a few hundred Hertz. There is now considerable work to develop DSP algorithms for analysis and classification, and compression of sampled biomedical signals [48]. IEEE Transactions on Biomedical Engineering is a good source for further information. See also *ECG, EMG, Evoked Potentials*.

Bipolar (1): A type of integrated circuit that uses NPN or PNP bipolar transistors in its construction [45].

Bipolar (2): *Bipolar* refers to the type of signalling method used for digital data transmission, in which either the marks or the spaces are indicated by successively alternating positive and negative polarities. See also *Non-return to Zero, Polar.*

Bit: A single binary digit; a 0 (a space) or 1 (a mark).

Bit Error Rate (BER): The fraction of bits in error occurring in a received bit stream. BER is calculated as the average number of bits in error, divided by the total number of bits in a given binary digit data stream. See also *BER vs. S/N Test.*

Bit Reverse Addressing: Due to the nature of the FFT algorithm it is often required to access data from memory in a non-arithmetic sequence (i.e. not 0,1,2, etc.) but in a sequence which is generated by reversing the address bits. As this type of addressing is very common to a DSP processor computing FFTs, this special addressing mode is available in some DSP processors to make programming easier, and algorithm execution faster. See also *Decimation-in-Time, Decimation-in-Frequency, FFT*.

Bit Serial Multiplier: See *Parallel Multiplier*.

Bitstream: *Bitstream* (Philips technology) DACs use sigma-delta technology to produce low cost and precise digital to analog converters. See *Sigma Delta*.

Blackmann Window: See *Window.*

Blackmann-harris Window: See *Window.*

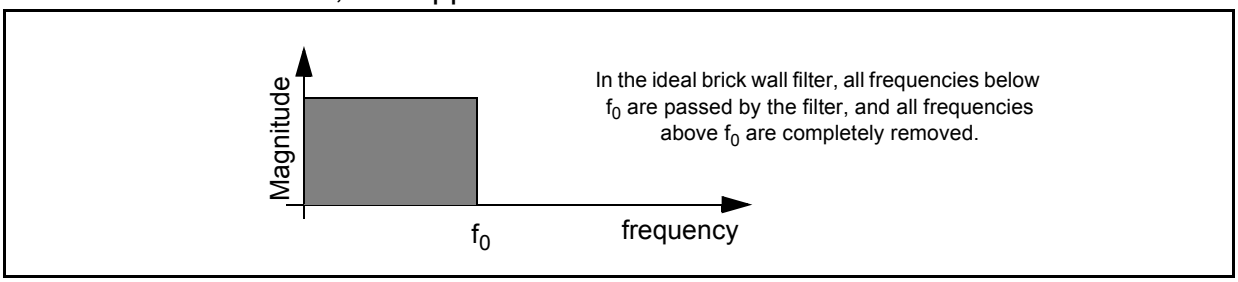
Blue Book: Shorthand name for the ITU-T regulations published in 1988 in 20 volumes and 61 Fascicles with a blue cover! (The ITU were known as the CCITT in 1988.) See also *International Telecommunication Union*, *Red Book, Standards*.

Board: See *DSP Board*.

Bounded: When the upper and lower values of specific parameters of a signal (or function) are known, or can be calculated or inferred from prior knowledge, then that parameter is said to be *bounded*.

Boxcar Filter: See *Moving Average.*

Brick Wall Filter: This is a filter having a frequency response that falls off to zero with infinite slope at some specified frequency. Although such filters are desirable in various DSP applications a true *brick wall filter* does not exist, and approximations with tolerable errors must be made.

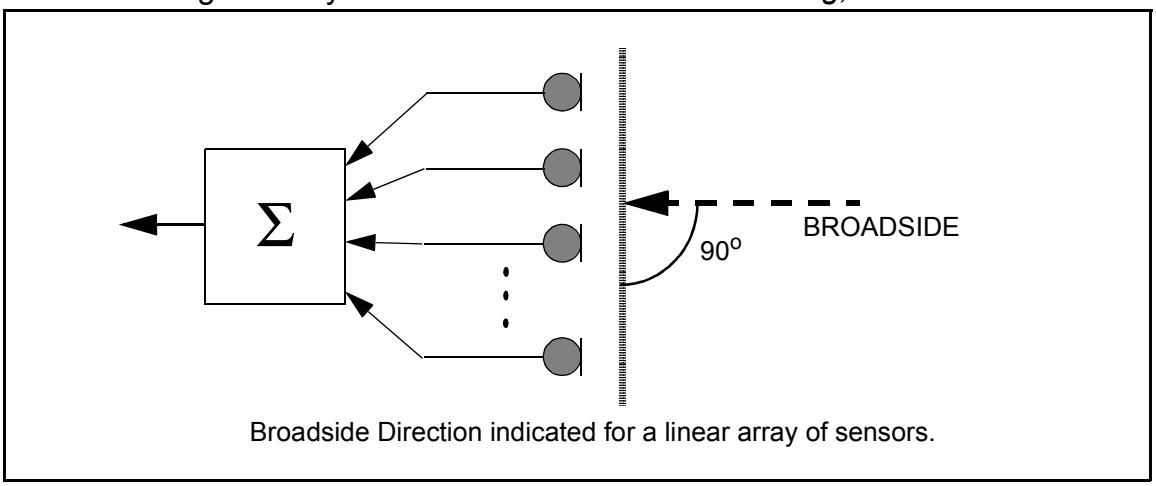

Magnitude
f_0
frequency
In the ideal brick wall filter, all frequencies below f_0 are passed by the filter, and all frequencies above f_0 are completely removed.

Broadband: See *Wideband*.

Broadband Hiss: If a speech or music signal has a relatively low level of superimposed white noise then this is referred to as *broadband hiss*. The term hiss is onomatopoeic -- the prolonged sound of the "*ss*'s" gives a good simulation of the phenomenon. See also *Dithering, White Noise*.

Broadband Integrated Digital Services Network (BISDN): Generally, BISDN refers to the information infrastructure provided by communications companies and institutions. The term BISDN evolved from the Integrated Services Digital Network (ISDN) to be a superset of the hardware and protocols provided by a previously adequate network infrastructure.

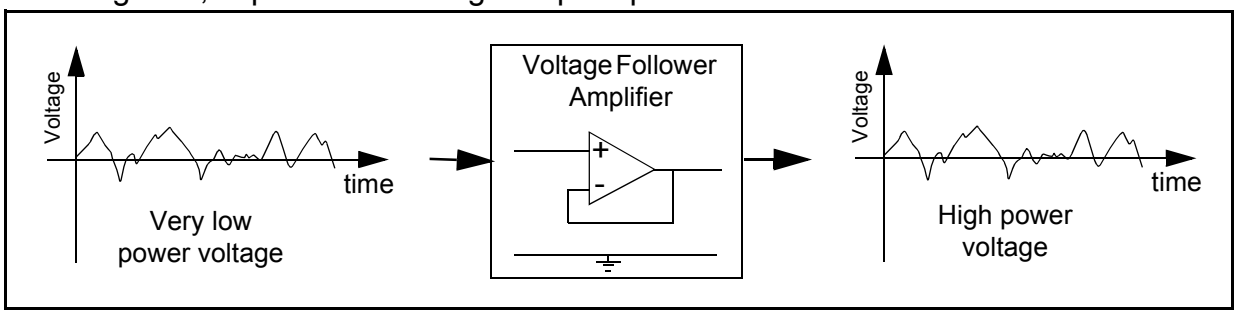
Broadside: A beamformer configuration in which the desired signal is located at right angles to the line or plane containing an array of sensors. See also *Beamforming, Endfire*.



Broadside Direction indicated for a linear array of sensors.

Buffer: Usually an area of memory used to store data temporarily. For example a large stream of sampled data is *buffered* in memory as 1000 sample chunks prior to digital signal processing. Buffers are also used in data communications to compensate for changes in the rate of data flow (e.g., rate fluctuations due to data compression algorithms).

Buffer Amplifier: An amplifier with a high input impedance and low output impedance that has a voltage gain of one. If, for example, a sensor outputs an analog voltage that is of the appropriate magnitude to input to an ADC, but it cannot deliver or sink enough current, then a buffer amplifier can be used prior to the ADC converter. The simplest form of buffer amplifier to build is a voltage follower with gain 1, implemented using an op-amp.

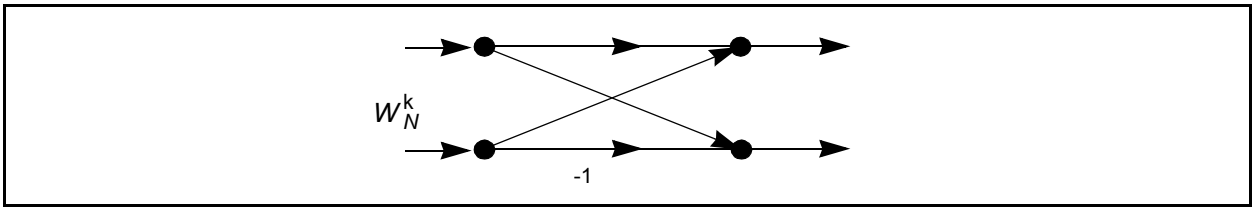


Burst Errors: When a large number of bits are incorrect in a relatively short segment of data bits then a *burst error* has occurred. In *burst errors* the average bit error rate is greatly exceeded by multiple bit errors. When the number of bits in error is very high then non-interleaved error correction schemes are unlikely to be successful and retransmission of the data may be required. See also *Channel Coding, Interleaving, Cross-Interleaved Reed-Solomon Coding.*

Bus: The generic name given to a set of wires used to transmit digital information from one point to another. A *bus* can be on-chip or off-chip. See also *DSP Processor.*

Busy Tone: Tones at 480 Hz and 620 Hz make up the busy tone for telephone systems.

Butterfly: The name given to the signal flow graph (SFG) element which can be used as a basic computational element to construct an *N* point fast Fourier transform (FFT) computation. See also *FFT*.



Butterworth Filter: See *Filters*.

Byte: 8 *bits*. 2 *nibbles*.

C

Cable (1): One or more conductors (such as copper wire) or other transmission media (such as optical fiber) within a protective sheath (usually plastic) to allow the efficient propagation of signals.

Cable (2): A generic name for cable TV systems using coaxial *cable* and/or optical fibers to transmit signals. *Cable* was first introduced into areas of the USA where geographical features prevented normal terrestrial TV reception. Within a few years of its introduction it proved so popular, flexible and reliable that *cable* became widely available all over the USA. Currently cable companies are involved in developing digital broadcast systems, and interactive TV viewing features.

Cache: A useful means of keeping often used data or information handy, a *cache* is simply a buffer of memory whose contents are updated according to an algorithm that is designed to minimize the number of data accesses that require looking beyond the *cache* memory. Both hardware and software implementations of the *cache* algorithms are common in DSP systems.

Call Progress Detection (CPD): A technique for monitoring the connection status during initiation of a telephone call by detecting the presence of call progress signalling tones such as the dialing tone, or the engaged (busy) signals as commonly found in the telephone network.

Carrier Board: A printed circuit board that can host a number of daughter modules providing facilities such as a DSP processor, memory, and I/O channels. A *carrier board* without daughter modules has no real functionality. See also *DSP Board, DSP Processor*.

Carry Look-Ahead Adder: See entry for *Parallel Adder*.

Cassette Tape: See *Compact Cassette Tape*.

Cauchy-Schwartz Inequality: See *Vector Properties - Cauchy-Schwartz*.

Causal: A signal produced by a real device or system is said to be *causal*. If a signal generating device is turned on at time, $t = t_0$, then the resultant signal produced exists only after time, $t = t_0$:

$$y(t) = \begin{cases} x(t) \text{ if } t \geq t_0 \\ 0 \quad \text{if } t < t_0 \end{cases} \tag{44}$$

SIgnals that are not *causal* are said to be non-causal. Although in the real world all signals are necessarily *causal*, from a mathematical viewpoint non-causal signals can be useful for the analysis of signals and systems.

Central Processing Unit (CPU): The part of the processor that performs that actual processing operations of addition, multiplications, comparison etc. The size of the arithmetic in the CPU usually defines the processor wordlength. For example the DSP56002 has a 24 bit CPU, meaning that it is a 24 bit processor. Usually the CPU wordlength matches the data bus width. If a DSP processor is floating point, then the CPU will also be capable of floating point arithmetic. See also *DSP Processor*.

Channel: The generic name given to the transmission path of any signal, which usually changes the signal characteristics, e.g. a telephone *channel*.

Also used to mean the input or output port of a DSP system. For example a DSP board with two ADCs and one DACs would be described as a twin channel input, single *channel* output system.

Channel Coding: This refers to the coding of information data that introduces structured redundancy so that inevitable errors introduced by transmitting symbols over noisy channels will be correctable (or at least detectable) at the receiver. The simplest *channel codes* are single bit parity checks (a simple block code). Other, more involved block codes and convolutional codes exist. In block coding a block of *k* data bits are encoded into *n* code bits to yield a rate *k/n* code. Block codes tend to have large *k* and large *n.* In convolutional coding the coder maintains a memory of previous data bits and outputs *n* code bits for each *k* input bits (using not only the input data bits but also those data bits stored in the coder memory) to yield a rate *k/n* code. Convolutional codes tend to have small values of *k* and *n* with coding strength determined by the amount of memory in the coder. Block and/or convolutional coding techniques can be combined to produce very strong (often cross-interleaved) codes. See also *Source Coding, Interleaving, Cross-Interleaved Reed-Solomon Code.*

Characteristic Polynomial: In order to conveniently specify the code used for cyclic redundancy coding (CRC) or a pseudo random binary sequence, a *characteristic polynomial* is often referred to. For example, the divisor using in ITU-T V.41 error control is 10001000000100001 is easier to represent as:

$$X^{16} + X^{12} + X^5 + 1 \tag{45}$$

The index of each term in this polynomial indicates a 1 in the divisor (i.e. the divisor has 1's at positions 0, 1, 5, 12 and 16). See also *Pseudo-Random Binary Sequence*.

Chebyshev Filter: See *Filters*.

Character: Letter, number, punctuation or other symbol. *Characters* are the basic unit of textual information. In DSP enabled data communication most *characters* are represented by ASCII codes. See also *ASCII, EBCDIC*.

Chip: Integrated Circuit.

Chip Interval: The clocking period of a pseudo random binary sequence generator. See also *Pseudo Random Binary Sequence Generator*.

Cholesky Decomposition: See *Matrix Decompositions - Cholesky*.

Chorus: A music effect where a delayed, and perhaps low pass filtered version of a signal is added to the original signal to create a *chorus* or echoic sound. See also *Music, Music Synthesis*.

Chromatic Scale: The complete set of 12 notes in one octave of the Western music scale is often referred to as the *chromatic scale*. Each adjacent note in the chromatic scale differs by one semitone, which corresponds to multiplying the lower frequency by the twelfth root of 2, i.e. $2^{1/12} = 1.0594631\ldots$. The *chromatic scale* is also known as the equitempered scale. See also *Western Music Scale*.

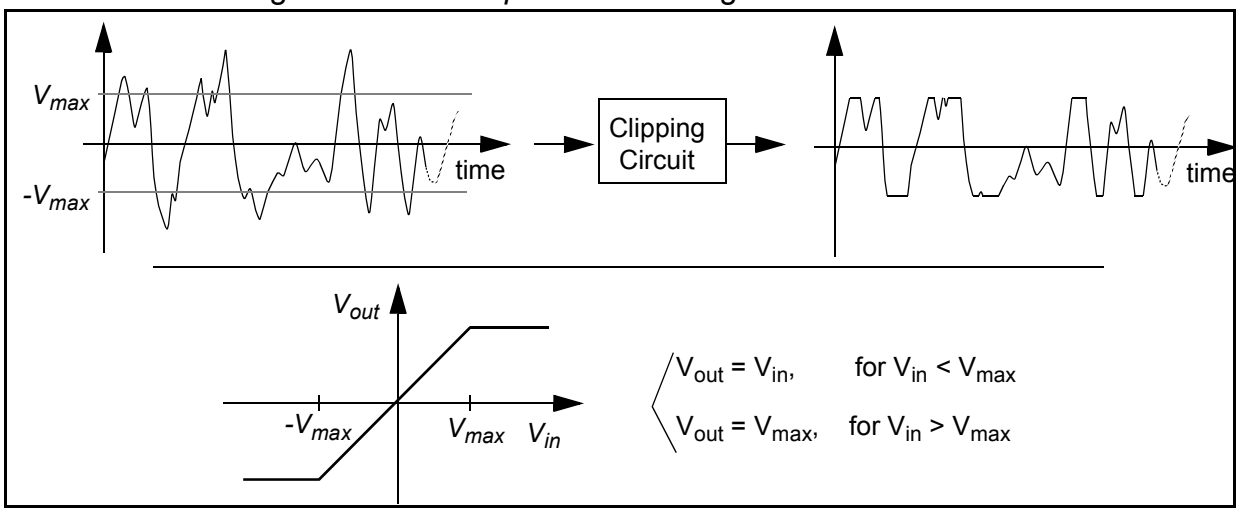
Circulant Matrix: See *Matrix Structured - Circulant.*

Circular Buffers: This is a effectively a programming concept that allows fast and efficient implementation of shift registers in memory to allow convolutions, FIR filters, and correlations to be executed with a minimum of data movement as each new data sample arrives. Modulo registers, and indirect pointers facilitate *circular buffers*.

Circular Reasoning: See *Reasoning, Circular.*

CISC: Complex Instruction Set Computer (see *RISC* definition)

Clipping: The *nonlinear* process whereby the value of an input voltage is limited to some maximum and minimum value. An analog signal with a magnitude larger than the upper and lower bounds $\pm V_{max}$ of an ADC chip, will be clipped. Any voltage above V_{max} will be *clipped* and the information lost. *Clipping* effects frequently occur in amplifiers when the amplification of the input signal results in a value greater than the *power rail* voltages.



Clock: A device which produces a periodic square wave that can be used to synchronize a DSP system. Current technology can produce extremely accurate *clocks* into the MHz range of frequencies.

Clock Jitter: If the clock edges of a clock vary in time about their nominal position in a stochastic manner, then this is *clock jitter*. In ADCs and DACs *clock jitter* will manifest itself as a raising of the noise floor [78]. See also *Quantization Noise*.

CMOS (Complimentary Metal Oxide Silicon): The (power efficient) integration technology used to fabricate most DSP processors.

Cochlea: The mechanics of the *cochlea* convert the vibrations from the bones of the middle ear (i.e., the ossicles, often called the hammer, anvil and stirrup) into excitation of the acoustic nerve endings. This excitation is perceived as sound by the brain. See also *Ear*.

Codebook Coding: A technique for data compression based on signal prediction. The compressed estimate is derived by finding the model that most closely matches the signal based on previous signals. Only the error between the selected model and the actual signal needs to be transmitted. For many types of signal this provides excellent data compression since, provided the codebook is sufficiently large, errors will be small. See also *Compression.*

Codec: A *CO*der and *DEC*oder. Often used to describe a matched pair of A/D and D/A converters on a single CODEC chip usually with logarithmic quantizers (A-law for Europe and μ-law for the USA.)

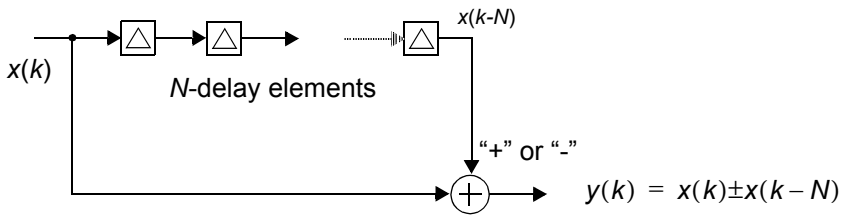
Coded Excited Linear Prediction Vocoders (CELP): The *CELP* vocoder is a speech encoding scheme that can offer good quality speech as relatively low bit rates (4.8kbits/sec) [133]. The drawback is that this vocoder scheme has a very high computational requirement. *CELP* is essentially a vector quantization scheme using a codebook at both analyzer and synthesizer. Using *CELP* a 200Mbyte hard disk drive could store close to 100 hours of digitized speech. See also *Compression*.

Coherent: Refers to a detection or demodulation technique that exploits and requires knowledge of the phase of the carrier signal. *Incoherent* or *Noncoherent* refers to techniques that ignore or do not require this phase information.

Color Subsampling: A technique widely used in video compression algorithms such as *MPEG1*. Color subsampling exploits the fact that the eye is less sensitive to the color (or chrominance) part of an image compared to the luminance part. Since the eye is not as sensitive to changes in color in a small neighborhood of a given pixel, this information is subsampled by a factor of two in each dimension. This subsampling results in one-fourth of the number of chrominance pixels (for each of the two chrominance fields) as are used for the luminance field (or brightness). See also *Moving Picture Experts Group*.

Column Vector: See *Vector*.

Comb Filter: A comb digital filter is so called because the magnitude frequency response is periodic and resembles that of a comb. (It is worth noting that the term “comb filter” is not always used consistently in the DSP community.) Comb filters are very simple to implement either as an FIR filter type structure where all weights are either 1, or 0, or as single pole IIR filters. Consider a simple FIR comb filter:



The simple comb filter can be viewed as an FIR filter where the first and last filter weights are 1, and all other weights are zero. The comb filter can be implemented with only a shift register, and an adder; multipliers are not required. If the two samples are added then the comb filter has a linear gain factor of 2 (i.e 6 dB) at 0 Hz (DC) thus in some sense giving a low pass characteristics at low frequencies. And if they are subtracted the filter has a gain of 0 giving in some sense a band stop filter characteristic at low frequencies.

The transfer function for the FIR comb filters can be found as:

$$\begin{aligned} Y(z) &= X(z) \pm z^{-N} X(z) = (1 \pm z^{-N}) X(z) \\ \Rightarrow H(z) &= \frac{Y(z)}{X(z)} = (1 \pm z^{-N}) \end{aligned} \tag{46}$$

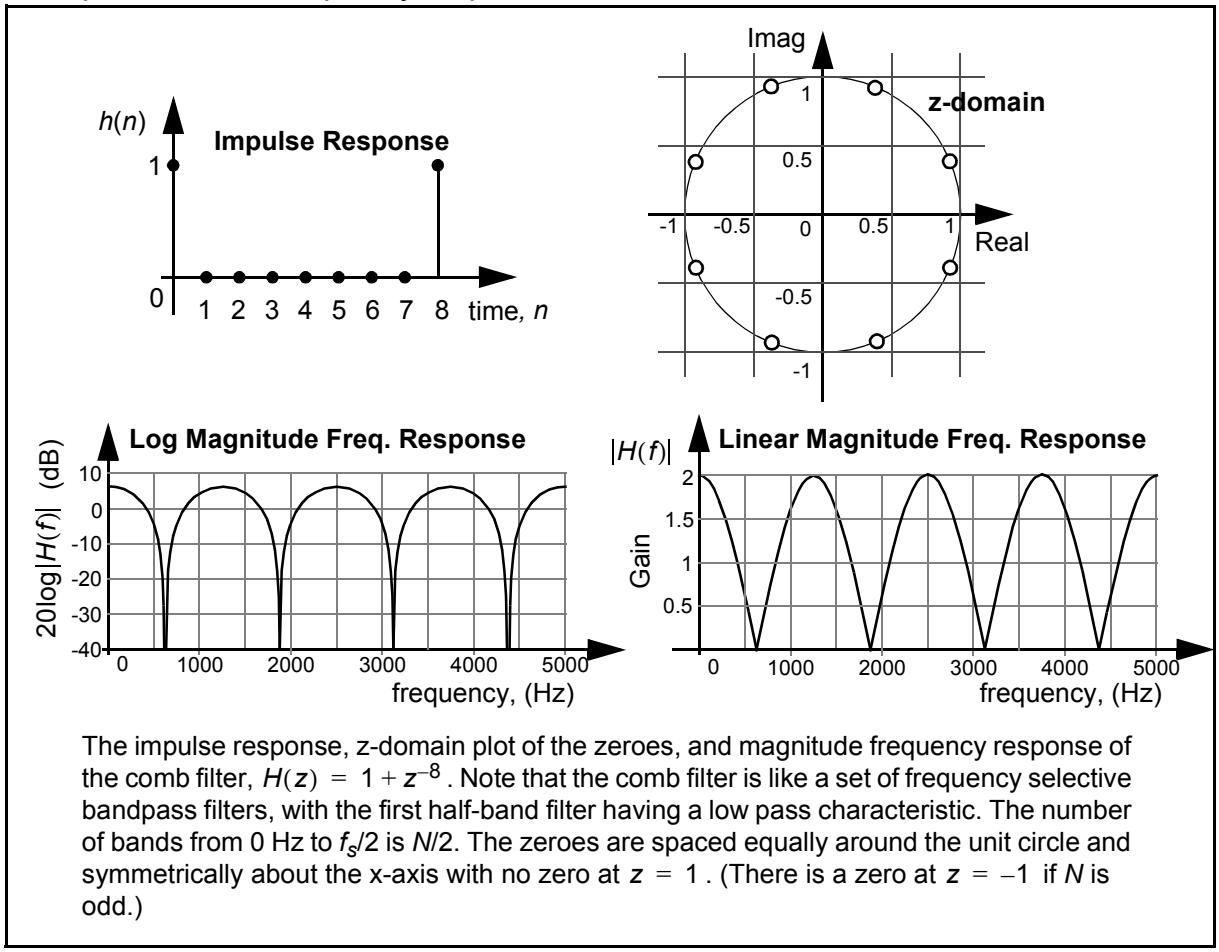
The zeroes of the comb filter, are the N roots of the z-domain polynomial $1 \pm z^{-N}$: Therefore for the case where the samples are subtracted:

$$\begin{aligned} 1 - z^{-N} &= 0 \\ \Rightarrow z_n &= \sqrt[N]{1} \text{ where } n = 0 \ldots N-1 \\ \Rightarrow z_n &= \sqrt[N]{e^{j2\pi n}} \text{ noting } e^{j2\pi n} = 1 \\ \Rightarrow z_n &= e^{\frac{j2\pi n}{N}} \end{aligned} \tag{47}$$

And for the case where the samples are added:

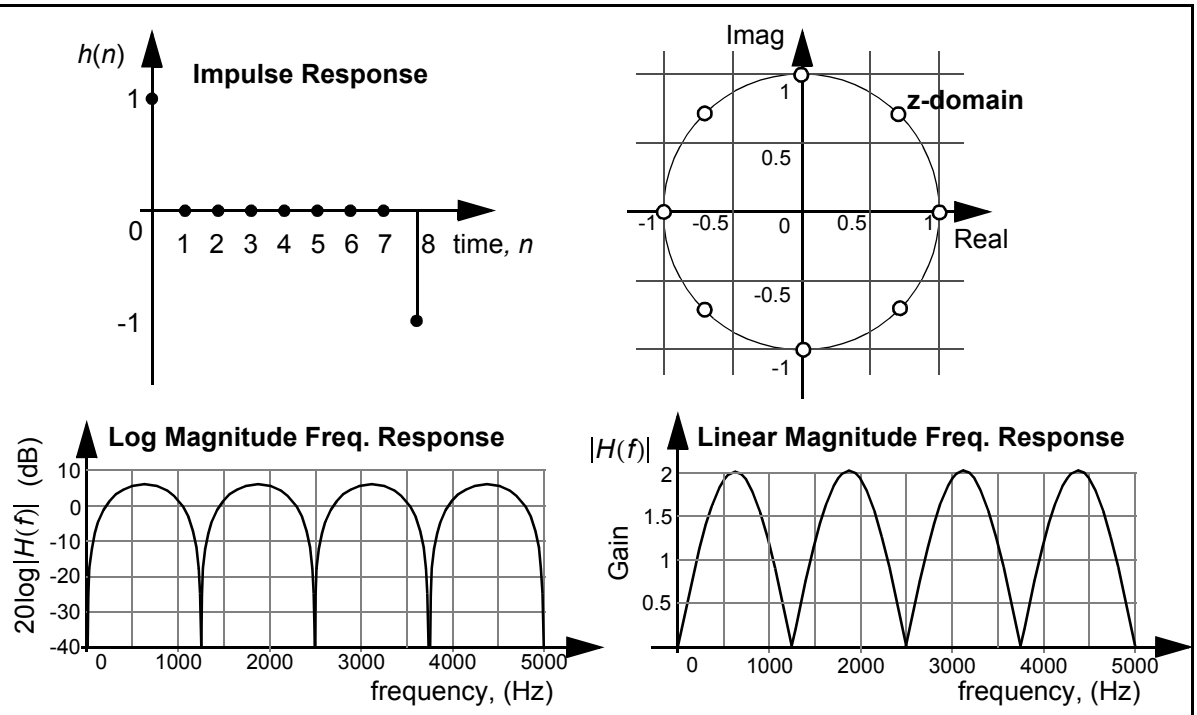
$$\begin{aligned} 1 + z^{-N} &= 0 \\ \Rightarrow z_n &= \sqrt[N]{-1} \text{ where } n = 0 \ldots N-1 \\ \Rightarrow z_n &= \sqrt[N]{e^{j2\pi\left(n+\frac{1}{2}\right)}} \text{ noting } e^{j2\pi\left(n+\frac{1}{2}\right)} = -1 \\ \Rightarrow z_n &= e^{\frac{j2\pi\left(n+\frac{1}{2}\right)}{N}} \end{aligned} \tag{48}$$

As an example, consider a comb filter $H(z) = 1 + z^{-8}$ and a sampling rate of $f_s = 10000$ Hz. The impulse response, $h(n)$, frequency response, $H(f)$, and zeroes of the filter can be illustrated as:



The impulse response, z-domain plot of the zeroes, and magnitude frequency response of the comb filter, $H(z) = 1 + z^{-8}$. Note that the comb filter is like a set of frequency selective bandpass filters, with the first half-band filter having a low pass characteristic. The number of bands from 0 Hz to $f_s/2$ is $N/2$. The zeroes are spaced equally around the unit circle and symmetrically about the x-axis with no zero at $z = 1$. (There is a zero at $z = -1$ if N is odd.)

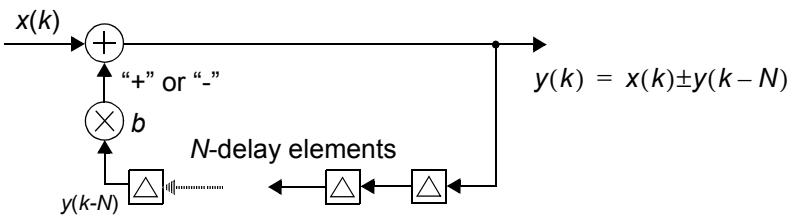
For the comb filter $H(z) = 1 - z^{-8}$ and a sampling rate of $f_s = 10000$ Hz. The impulse response, $h(n)$, frequency response, $H(f)$, and zeroes of the filter are:



The impulse response, z-domain plot of the zeroes, and magnitude frequency response of the comb filter, $H(z) = 1 - z^{-8}$. The zeroes are spaced equally around the unit circle and symmetrically about the x-axis. There is a zero at $z = 1$. There is not a zero at $z = -1$ if N is odd.

FIR comb filters have linear phase and are unconditionally stable (as are all FIR filters). For more information on unconditional stability and linear phase see entry for *Finite Impulse response Filters.*

Another type of comb filter magnitude frequency response can be produced from a single pole IIR filter:

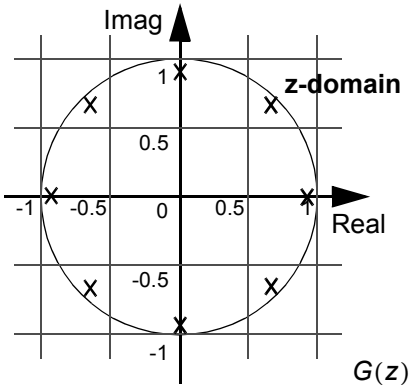


A single pole IIR comb filter. The closer the weight value *b* is to 1, then the sharper the teeth of the *comb filter* in the frequency domain (see below). *b* is of course less than 1, or instability results.

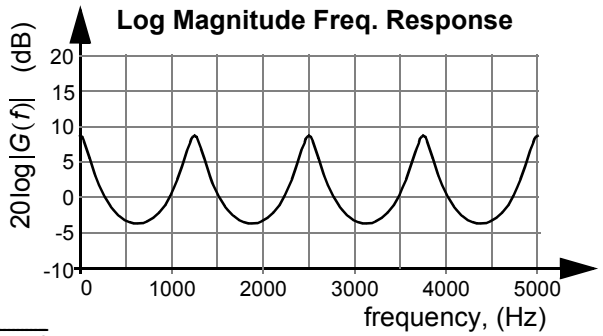
This type of comb filter is often used in music synthesis and for soundfield processing [43]. Unlike the FIR comb filter note that this comb filter does require at least one multiplication operation. Consider the difference equation of the above single pole IIR *comb filter:*

$$y(k) = x(k) \pm b y(k-N)$$
$$\Rightarrow G(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 \pm b z^{-N}} \qquad (49)$$

For a sampling rate of $f_s = 10000$ Hz, $N = 8$ and $b = 0.6$ the impulse response $g(n)$, the frequency response, $G(f)$, and poles of the filter are:

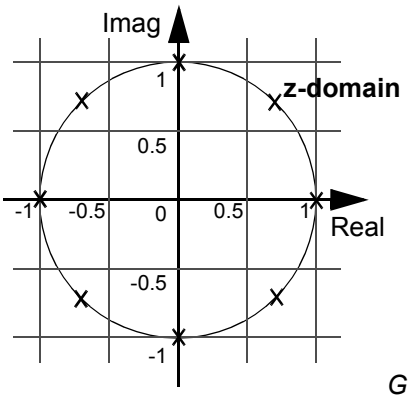


$$G(z) = \frac{1}{1 - 0.6z^{-8}}$$

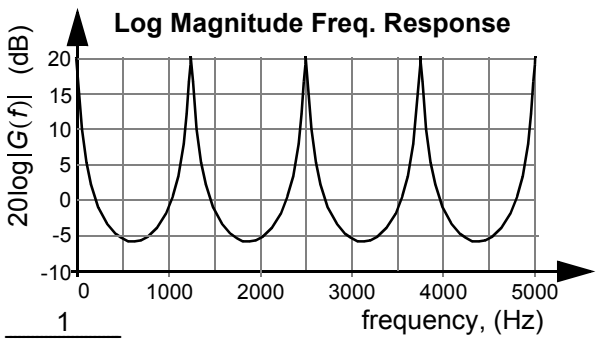


The z-domain plot of the filter poles and magnitude frequency response of one pole comb filter. The poles are inside the unit circle and lie on a circle of radius $0.6^{1/8} = 0.938\ldots$ As the feedback weight value, b, is decreased (closer to 0), then the poles move away from the unit circle towards the origin, and the peaks of the magnitude frequency response become less sharp and provide less gain.

Increasing the feedback weight, b, to be very close to 1, the "teeth" of the filter become sharper and the gain increases:



$$G(z) = \frac{1}{1 - 0.9z^{-8}}$$



The z-domain plot of the filter poles and magnitude frequency response of a one pole comb filter. The poles are just inside the unit circle and lie on a circle of radius $0.9^{1/8} = 0.987\ldots$

Of course if b is increased such that $b \geq 1$ then the filter is unstable.

The IIR comb filter is mainly used in computer music [43] for simulation of musical instruments and in soundfield processing [33] to simulate reverberation.

Finally it is worth noting again that the term "*comb filter*" is used by some to refer to the single pole IIR comb filter described above, and the term "inverse *comb filter*" to the FIR *comb filter* both

described above. Other authors refer to both as comb filters. The uniting feature however of all comb filters is the periodic (comb like) magnitude frequency response. See also *Digital Filter, Finite Impulse Response Filter, Finite Impulse Response Filter-Linear Phase, Infinite Impulse Response Filter, Moving Average Filter*. .

Comité Consultatif International Télégraphique et Téléphonique (CCITT): The English translation of this French name is the *International Consultative Committee on Telegraphy and Telecommunication* and is now known as the ITU-T committee. The ITU-T (formerly CCITT) is an advisory committee to the International Telecommunications Union (ITU) whose recommendations covering telephony and telegraphy have international influence among telecommunications engineers and manufacturers. See also *International Telecommunication Union, ITU-T*.

Comité Consultatif International Radiocommunication (CCIR): The English translation of this French names is the *International Consultative Committee on Radiocommunication* and is now known as the ITU-R committee. The ITU-R (formerly CCIR) is an advisory committee to the International Telecommunications Union (ITU) whose recommendations covering radiocommunications have international influence among radio engineers and manufacturers. See also *International Telecommunication Union, ITU-R*.

Comité Européen de Normalisation Electrotechnique (CENELEC): CENELEC is the European Committee for Electrotechnical Standardization. They provide European standards over a wide range of electrotechnology. CENELEC has drawn up an agreement with European Telecommunications Standards Institute (ETSI) to study telecommunications, information technology and broadcasting. See also *European Telecommunications Standards Institute, International Telecommunication Union, International Organisation for Standards, Standards*.

Common Intermediate Format (CIF): The *CIF* image format has 288 lines by 360 pixels/line of luminance information and 144 x 180 of chrominance information and is used in the |TU-T H261 digital video recommendation. A reduced version of CIF called quarter CIF (QCIF) is also defined in H261. The choice between CIF and QCIF depends on channel bandwidth and desired quality. See also *H-series Recommendations, International Telecommunication Union, Quarter Common Intermediate Format.*

Compact Cassette Tape: *Compact cassette tapes* were first introduced in the 1960s for convenient home recording and audio replay. By the end of the 1970s *compact cassette* was one of the key formats for the reproduction of music. Currently available compact cassettes afford a "good" response of about 65dB dynamic range from 100Hz to 12000Hz or better. *Compact cassette* outlived vinyl records, and is still a very popular format for music particularly in automobile audio systems. In the early 1990s DCC (Digital Compact Cassette) was introduced which had backwards compatibility with *compact cassette*. See also *Digital Compact Cassette*.

Compact Disc (CD): The digital audio system that stores two channels (stereo) of 16-bit music sampled at 44.1kHz. Current CDs allow almost 70 minutes of music to be stored on one disc (without compression). This is equivalent to a total of

$$2 \times 44100 \times 70 \times 60 \times 16 \; = \; 5927040000 \text{ bits of information.} \qquad (50)$$

CDs use cross-interleaved Reed-Solomon coding for error protection. See also *Digital Audio Tape (DAT), Red Book, Cross-Interleaved Reed-Solomon Coding.*

Compact Disc-Analogue Records Debate: Given that the bandwidth of hi-fidelity digital audio systems is up to 22.05kHz for compact disc (CD) and 24kHz for DAT it would appear that the full range of hearing is more than covered. However this is one of the key issues of the *CD-analogue records* debate. The argument of some analog purists is that although humans cannot perceive individual tones above 20kHz, when listening to musical instruments which produce harmonic frequencies above the human range of hearing these high frequencies are perceived in some "collective" fashion. This adds to the perception of live as opposed to recorded music; the debate will probably continue into the next century. See also *Compact Disc, Frequency Range of Hearing, Threshold of Hearing*.

Compact Disc ROM (CD-ROM): As well as music, CDs can be used to store general purpose computer data, or even video. Thus the disk acts like a *Read Only Memory* (ROM).

Companders: *Com*pressor and ex*pander* (*compander*) systems are used to improve the SNR of channels. Such systems initially attenuate high level signal components and amplify low level signals (compression). When the signal is received the lower level signals appear at the receiving end at a level above the channel noise, and when expansion (the inverse of the compression function) is applied an improved signal to noise ratio is maintained. In addition, the original signal is preserved by the inverse relationship between the compression and expansion functions. In the absence of quantization, *companders* provide two inverse 1-1 mappings that allow the original signal to be recovered exactly. Quantization introduces an irreversible distortion, of course, that does not allow exact recovery of the original signal. See also *A-law and* μ*-law.*

Comparator: A device which compares two inputs, and gives an output indicating which input was the largest.

Complex Base: In everyday life base 10 (decimal) is used for numerical manipulation, and inside computers base 2 (binary) is used. When complex numbers are manipulated inside a DSP processor, the real parts and complex parts are treated separately. Therefore to perform a complex multiplication of:

$$(a+jb)(c+jd) = (ac-bd)+j(ad+bc) \tag{51}$$

where 16 bit numbers are used to represent a, b, c, and d will require four separate real number multiplications and two additions. Therefore an interesting alternative (although not used in an practice to the authors' knowledge) is to use the *complex base* $(1+j)$, where only the digits 0, 1, and *j* are used. Setting up a table of the powers of this base gives:

$(1+j)^4$ **-4**	$(1+j)^3$ **-2+2*j***	$(1+j)^2$ **2*j***	$(1+j)^1$ **1+*j***	$(1+j)^0$ **1**	**Complex Decimal**
0	0	1	1	0	1 + 3*j*
0	0	0	*j*	0	-1 - *j*
0	*j*	1	1	1	*j*
1	0	1	1	0	-3 + 3*j*

Numbers in the complex base $(1+j)$ can then be arithmetically manipulated (addition, subtraction, multiplication) although this is not as straightforward as for binary!

Complex Conjugate: A complex number is conjugated by negating the complex part of the number. The *complex conjugate* is often denoted by a "*". For example, if $a = 5 + 7j$, then $a^* = 5 - 7j$. (A complex number and its conjugate are often called a conjugate pair.) Note that the product of *aa** is always a real number:

$$aa^* = (5 + 7j)(5 - 7j) = 25 + 35j - 35j + 49 = 25 + 49 = 74 \tag{52}$$

and can clearly be calculated by summing the squares of the real and complex parts. (Taking the square root of the product aa^* is often referred to as the magnitude of a complex number.) The conjugate of a complex number expressed as complex exponential is obtained by negating the exponential power:

$$(e^{j\omega})^* = e^{-j\omega} \tag{53}$$

This can be easily seen by noting that:

$$e^{j\omega} = \cos\omega + j\sin\omega \text{ , and} \tag{54}$$

$$e^{-j\omega} = \cos(-\omega) + j\sin(-\omega) = \cos\omega - j\sin\omega \tag{55}$$

given that cosine is an even function, and sine is an odd function. Therefore:

$$e^{j\omega}e^{-j\omega} = e^0 = \cos^2\omega + \sin^2\omega \tag{56}$$

A simple rule for taking a *complex conjugate* is: “replace any *j* by *-j* “. See also *Complex Numbers.*

Complex Conjugate Reciprocal: The *complex conjugate reciprocal* of a complex number is found by taking the reciprocal of the complex conjugate of the number. For example, if $z = a + bj$, then the *complex conjugate reciprocal* is:

$$\frac{1}{z^*} = \frac{1}{a - bj} = \frac{a + bj}{a^2 + b^2} \tag{57}$$

See also *Complex Numbers, Pole-Zero Flipping.*

Complex Exponential Functions: An exponent of a complex number times *t*, the time variable, provides a fundamental and ubiquitous signal type for linear systems analysis: the damped exponential. These signals describe many electrical and mechanical systems encountered in everyday life, like the suspension system for an automobile. See also *Damped Exponential*.

Complex LMS: See *LMS algorithm.*

Complex Numbers: A *complex number* contains both a real part and a complex part. The complex part is multiplied by the imaginary number *j*, where *j* is the square root of -1. (In other branches of applied mathematics *i* is usually used to represent the imaginary number, however in electrical engineering *j* is used because the letter *i* is used to denote electrical current.) For the *complex number*:

$$a + jb \tag{58}$$

a is the real part, where $a \in \Re$ ($\Re$ is the set of real numbers) and jb is the imaginary part, where $b \in \Re$. Complex arithmetic can be performed and the result expressed as a real part and imaginary part. For addition:

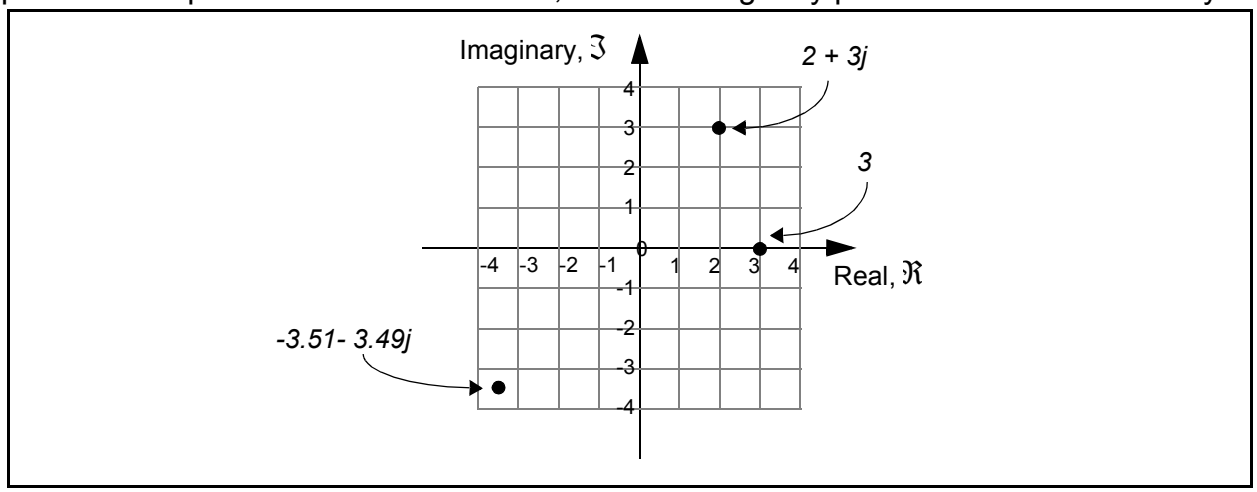
$$(a + jb) + (c + jd) = (a + c) + j(b + d) \tag{59}$$

and for multiplication:

$$(a + jb)(c + jd) = (ac - bd) + j(ad + bc) \tag{60}$$

Complex number notation is used to simplify Fourier analysis by allowing the expression of complex sinusoids using the complex exponential $e^{j\omega} = \cos\omega + j\sin\omega$. Also in DSP *complex numbers* represent a convenient way of representing a two dimensional space, for example in an adaptive beamformer (two dimensional space), or an adaptive decision feedback analyser where the in-phase component is a real number, and the quadrature phase component is a *complex number*. See also Complex Conjugate, Complex Sinusoid.

Complex Plane: The *complex plane* allows the representation of complex numbers by plotting the real part of a complex number on the x-axis, and the imaginary part of the number on the y-axis.



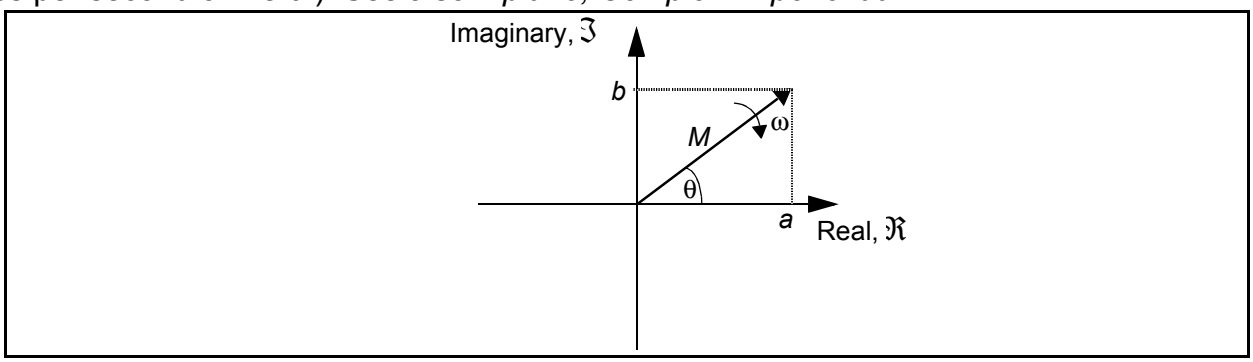
If a complex number is written as a complex exponential, then the *complex plane* plot can be interpreted as a phasor diagram, such that for the complex number $a + jb$:

$$a + jb = Me^{j\theta}, \tag{61}$$

where

$$\begin{aligned} M &= \sqrt{a^2 + b^2} \\ \theta &= \tan^{-1}\left(\frac{b}{a}\right) \end{aligned}. \tag{62}$$

If θ is a time dependent function such that $\theta = \omega t$, then the phasor will rotate in a counter-clockwise direction with angular frequency of ω radians per second (or $\omega/(2\pi)$ rotations per second, i.e., cycles per second or Hertz). See also *z-plane, Complex Exponential.*

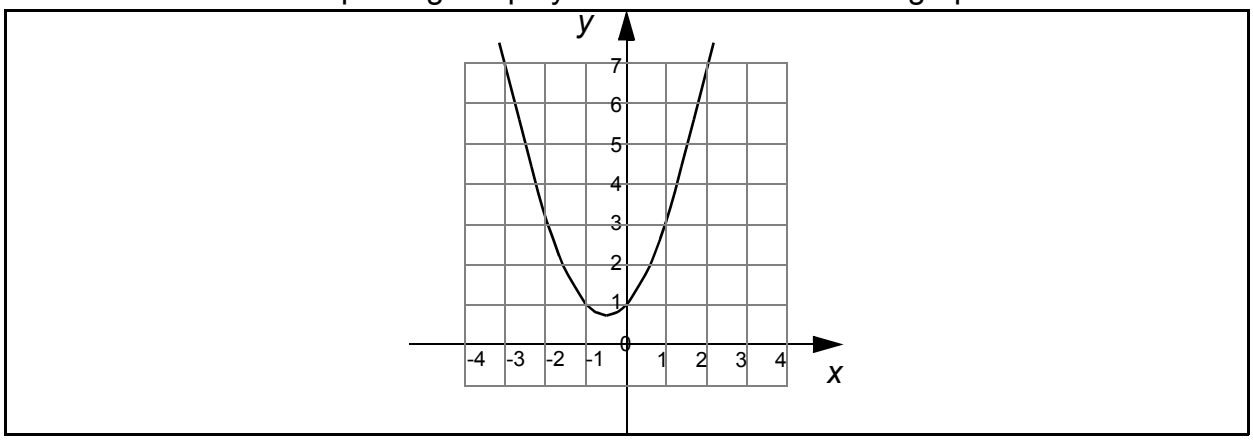


Conjugate Reciprocal: See *Complex Conjugate Reciprocal*.

Complex Roots: When the roots of a polynomials are calculated, if there is no real solution, then roots are said to be complex. As an example consider the following quadratic polynomial:

$$y = x^2 + x + 1 \tag{63}$$

The roots of this polynomial are when $y = 0$. Geometrically this is where the are where the graph of y cuts the x-axis. However plotting this polynomial it is clear that the graph does not cut the x-axis:



In this case the roots of the polynomial are not real. Using the quadratic formula we can calculated the roots as:

$$\begin{aligned} x &= \frac{-1 \pm\sqrt{1^2 - 4}}{2} \\ &= \frac{-1 \pm 3j}{2} \end{aligned} \tag{64}$$

and therefore:

$$x^2 + x + 1 = \left(x + \frac{1}{2} + \frac{\sqrt{3}}{2}j\right)\left(x + \frac{1}{2} - \frac{\sqrt{3}}{2}j\right) \tag{65}$$

This example indicates the fundamental utility of complex number systems. Note that the coefficients of the polynomial are real numbers. It is obvious from the plot of the polynomial that no real solution to $y(x) = 0$ exists. However, the solution does exist if we choose x from the larger set of complex numbers. In applications involving linear systems, these complex solutions provide a tremendous amount of information about the nature of the problem. Thus real world phenomena can be understood and predicted simply and accurately in a way not possible without the intuition provided by complex mathematics. See also *Poles, Zeroes*.

Complex Sinusoid: See *Damped Exponential*.

Compression: Over the last few years *compression* has emerged as one of the largest areas of real time DSP application for digital audio and video. The simple motivation is that the bandwidth required to transmit digital audio and video signals is considerably higher than the analogue transmission of the baseband analogue signal, and also that storage requirements for digital audio and video are very high. Therefore data rates are reduced by essentially reducing the data required to transmit of store a signal, while attempting to maintain the signal quality.

For example, the data rate of a stereo CD sampling at 44.1kHz, using 16 bit samples on stereo channels is:

$$\text{Data Rate} = 44100 \times 16 \times 2 = 1411200 \text{ bits/sec} \tag{66}$$

The often quoted CD transmission bandwidth (assuming binary signalling) is 1.5MHz. Compare this bandwidth with the equivalent analog bandwidth of around 30kHz for two 15kHz analog audio channels.

The storage requirements for 60 minutes of music in CD format are:

$$\text{CD Storage Requirement} = 44100 \times 2 \times 2 \times 60 \times 60 = 635 \text{ Mbytes/60 minutes} \tag{67}$$

In general therefore CD quality PCM audio is difficult to transmit, and storage requirements are very high. As discussed above, if the sampling rate is reduced or the data wordlength reduced, then of course the data rate will be reduced, however the audio quality will also be affected. Therefore there is a requirement for audio *compression* algorithms which will reduced the quantity of data, but will not reduce the perceived quality of the audio.

For telecommunications where speech is coded at 8kHz using, for example, 8 bit words the data rate is 64000 bits per second. The typical bandwidth of a telephone line is around 4000Hz, and therefore powerful *compression* algorithms are clearly necessary. Similarly teleconferencing systems require to compress speech coded at the higher rate of 16 kHz, and a video signal.

Ideally no information will be lost by a *compression* algorithm (i.e. lossless). However, the compression achievable with lossless techniques is typically quite limited. Therefore most audio compression techniques are lossy such that the aim of *compression* algorithm is to reduce the components of the signal that do not matter such as periods of silence, or sounds that will not be heard due to the psychoacoustic behaviour of the ear whereby loud sounds mask quieter ones.

For hi-fidelity audio the psychoacoustic or perceptual coding technique is now widely used to compress by factors between 2:1 and almost 12:1. Two recent music formats, the mini-disk and DCC (digital compact cassette) both use perceptual coding techniques and produce compress of 5:1 and 4:1 with virtually no (perceptual) degradation in the quality of the music. Digital audio

compression will continue to be a particularly large area of research and development over the next few years. Applications that will be enabled by real time DSP *compression* techniques include:

Telecommunications: Using toll-quality telephone lines to transmit compressed data and speech;

Digital Audio Broadcasting (DAB): DAB data rates must be as low as possible to minimise the required bandwidth;

Teleconferencing/Video-phones: Teleconferencing or videophones via telephone circuits and cellular telephone networks;

Local Video: Using image/video *compression* schemes medium quality video broadcast for organisations such as the police, hospitals etc are feasible over telephones, ISDN lines, or AM radio channels;

Audio Storage: If a signal is compressed by a factor of *M*, then the amount of data that can be stored on a particular medium increases by a factor of *M*.

The table below summarises a few of the well known audio *compression* techniques for both hi-fidelity audio and telecommunications. Currently there exist many different "standard" *compression* algorithms, and different algorithms have different performance attributes, some remaining proprietary to certain companies.

Algorithm	Compression Ratio	Bit/rate, kbits/sec	Audio Bandwidth (Hz)	Example Application
PASC	4:1	384	20kHz	DCC
Dolby AC-2	6:1	256	20kHz	Cinema Sound
MUSICAM	4:1 to 12:1	192 to 256	20kHz	Professional Audio
NICAM	2:1	676	16kHz	Stereo TV audio
ATRAC	5:1	307	20kHz	Mini-disc
ADPCM (G721)	8:5 to 4:1	16, 24, 32, 40	4kHz	Telecommunications
IS-54 VSELP	8:1	8	4kHz	Telecommunications
LD-CELP (G728)	4:1	8	4kHz	Telecommunications

Video *compression* schemes are also widely researched, developed and implemented. The best known schemes are Moving Picture Experts Group (MPEG) which is in fact both audio and video, and the ITU H-Series Recommendations (H261 etc). The Joint Photographic Experts Group (JPEG) standards and Joint Bi-level Image Group (JBIG) consider the compression of still images.

See also *Adaptive Differential Pulse Code Modulation, Adaptive Transform Acoustic Coding (ATRAC), Entropy Coding, Huffman Coding, Arithmetic Coding, Differential Pulse Code Modulation, Digital Compact Cassette, G-Series Recommendations, H-Series Recommendations, Joint Photographic Experts Group, MiniDisc, Moving Picture Experts Group, TransformCoding, Precision Adaptive Subband Coding, Run Length Encoding*.

Condition Code Register (CCR): The register inside a DSP processor which contains information on the result of the last instruction executed by the processor. Typically bits (or flags) in the CCR will indicate if the previous instruction had a zero result, positive result, overflow

occurred, the carry bit value. The CCR bits are then used to make conditional decisions (branching). The CCR is sometimes called the Status Register (SR). See also *DSP Processor*.

Condition Number: See *Matrix Properties - Condition Number.*

Conditioning: See *Signal Conditioning*.

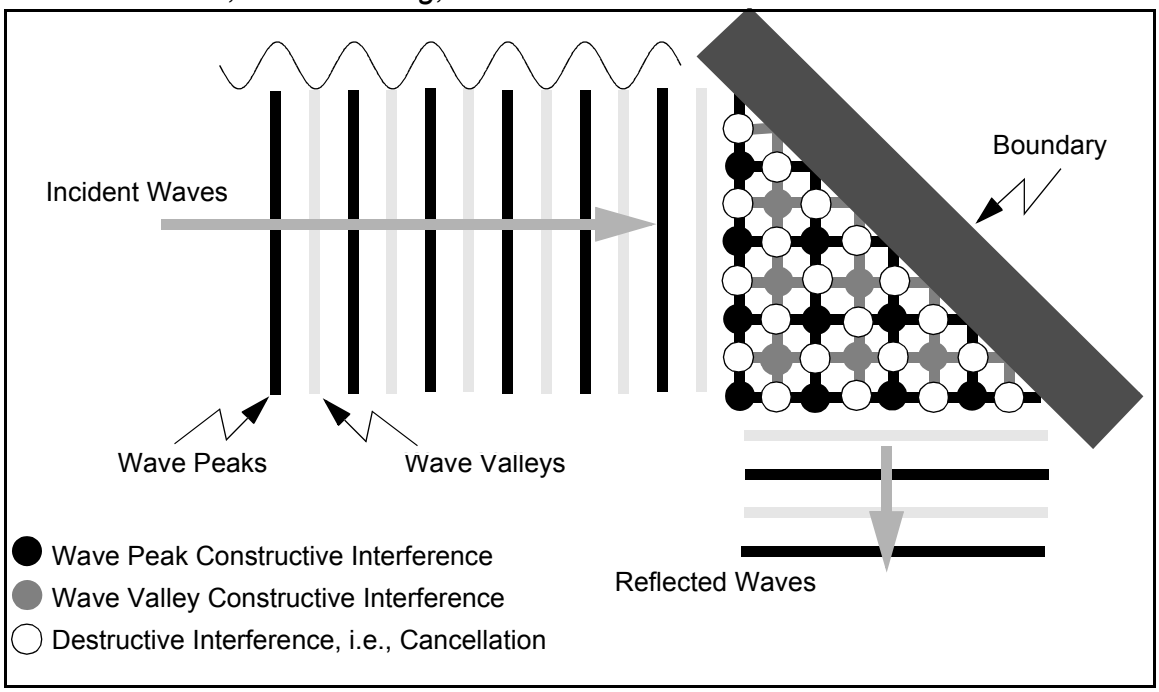
Conductive Hearing Loss: If there is a defect in the middle ear this can often reduce the transmission of sound to the inner ear [30]. A simple *conductive hearing loss* can be caused by as simple a problem as excessive wax in the ear. The audiogram of a person with a *conductive hearing loss* will often indicate that the hearing loss is relatively uniform over the hearing frequency range. In general a *conductive hearing loss* can be alleviated with an amplifying hearing aid. See also *Audiology, Audiometry, Ear, Hearing Aids, Hearing Impairment, Loudness Recruitment, Sensorineural Hearing Loss, Threshold of Hearing*.

Conjugate: See *Complex Conjugate*.

Conjugate Pair: See *Complex Conjugate*.

Conjugate Transpose: See *Matrix Properties - Hermitian Transpose*

Constructive Interference: The addition of two waveforms with nearly identical phase. *Constructive interference* is exploited to produce resonance in physical and electrical systems. *Constructive interference* is also responsible for energy peaks in diffraction patterns. See also *Destructive Interference, Beamforming, Diffraction.*



Continuous Phase Modulation (CPM): A type of modulation in which abrupt phase changes are avoided to reduce the bandwidth of the modulated signal. CPM requires increased decoder complexity. See also *Minimum Shift Keying, Viterbi Algorithm.*

Continuous Variable Slope Delta Modulator (CVSD): A speech compression technique that was used before ADPCM became popular and standardized by the ITU [133]. Although CVSD

generally produces lower quality speech it is less sensitive to transmission errors than ADPCM. See also *Compression, Delta Modulation*.

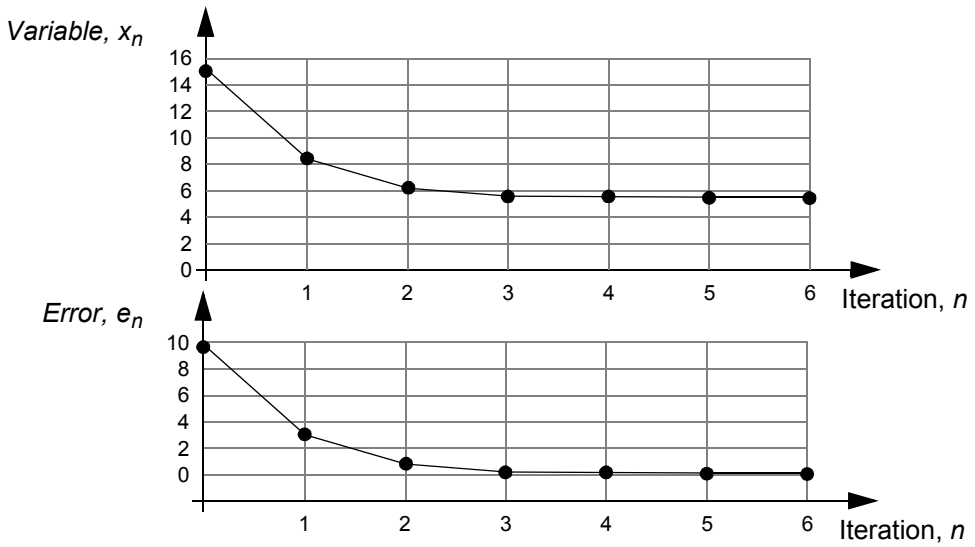
Control Bus: A collection of wires on a DSP processor used to transmit control information on chip and off chip. An example of control information is stating whether memory is to be read from, or written to. This would be indicated by the single $\overline{R}/W$ line. See also *DSP Processor*.

Convergence: Algorithms such as adaptive algorithms, are attempting to find a particular solution to a problem by *converging* or iterating to the correct solution. *Convergence* implies that the correct solution is found by continuously reducing the error between the current iterated value and the true solution. When the error is zero (or, more practically, relatively small), the algorithm is said to have *converged*. For example consider an algorithm which will update the value of a variable x_n to converge to the square root of a number, *a*. The iterative update is given by:

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right) \tag{68}$$

where the initial guess, x_0, is *a*/2. The error of $e_n = x_n - \sqrt{a}$ will reduce at each iteration, and *converge* to zero. Because most algorithms converge asymptotically, convergence is often stated to have occurred when a specified error quantity is less than a particular value.

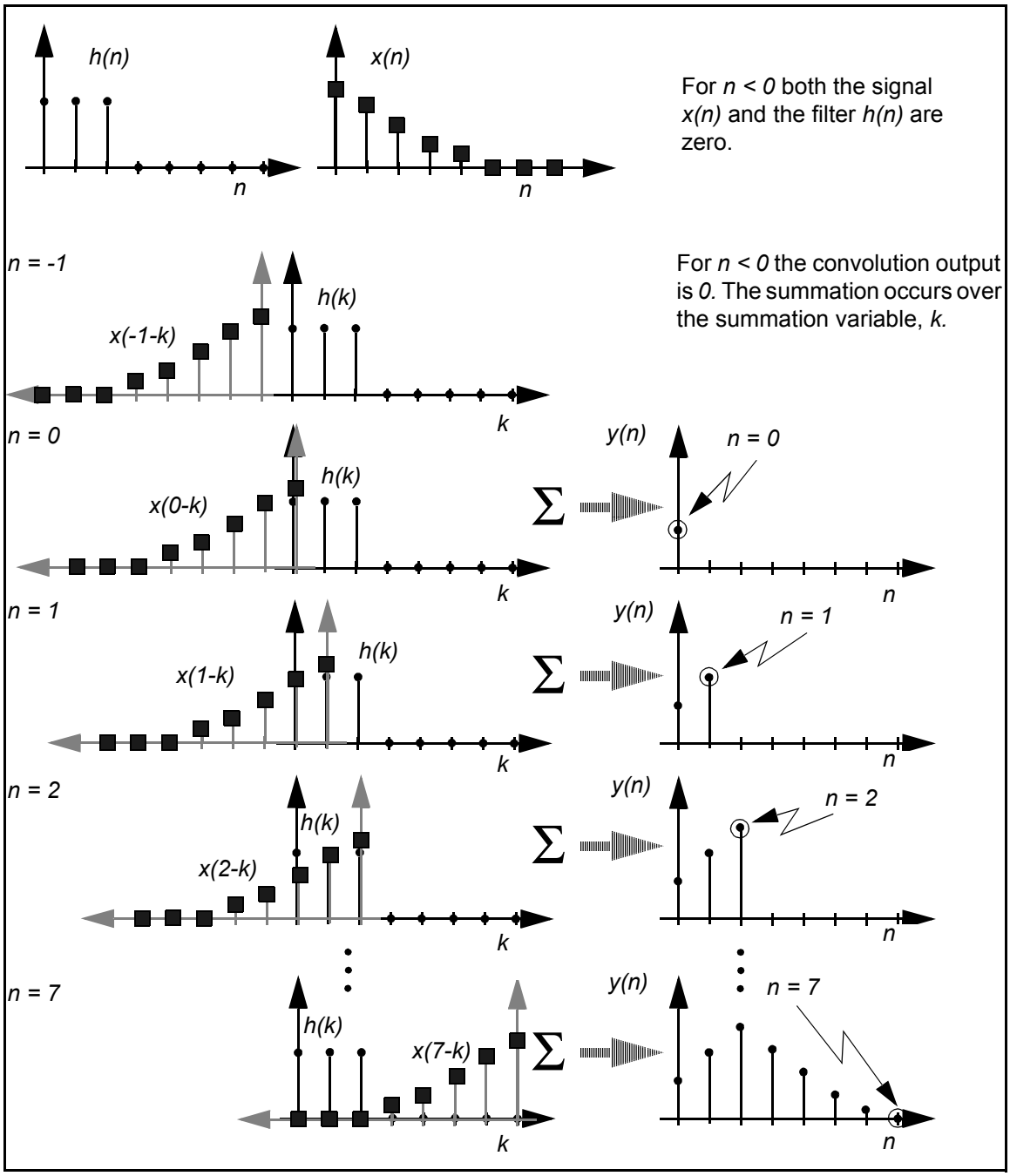
Finding the square root of a = 15, using an iterative algorithm to *converge* to the solution of $\sqrt{a}$ = 5.477 . Note that after only 6 iterations the algorithm has *converged* to within 0.03 of the correct answer



Another example is a system identification application using an adaptive LMS FIR filter to model an unknown system. *Convergence* is said to have occurred when the mean squared error between the output of the actual system and the modelled one (given the same input) is less than a certain value determined by the application designer. Algorithms that do not converge and perhaps **diverge**, are usually labelled as unstable. See also *Adaptive Signal Processing, Iterative Techniques*.

Convolution: When a signal is input to a particular linear system the impulse response of the system is *convolved* with the input signal to yield the output signal. For example, when a sampled speech signal is operated on by a digital low pass filter, then the output is formed from the convolution of the input signal and the impulse response of the low pass filter:

$$y(n) = h(n) \otimes x(n) = \sum_k h(k)x(n-k) \tag{69}$$



Cooley-Tukey: J.W. Cooley and J.W. Tukey published a noteworthy paper in 1965 highlighting that the discrete Fourier transform (DFT) could be computed in fewer computations by using the *fast Fourier transform* (FFT) [66]. Reference to the *Cooley-Tukey* algorithm usually means the FFT. See also *Fast Fourier Transform, Discrete Fourier Transform*.

Co-processor: Inside a PC, a processor that is additional to the general purpose processor (such as the Intel 80486) is described as a *co-processor* and will usually only perform demanding

computational tasks. For multi-media applications, DSP processors inside the PC to facilitate speech processing, video and communications are co-processors.

CORDIC: An arithmetic technique that can be used to calculate sin, cos, tan and trigonometrical values using only shift and adds of binary operands [25].

Core: All DSP applications require very fast MAC operations to be performed, however the algorithms to be implemented, and the necessary peripherals to input data, memory requirements, timers and on-chip CODEC requirements are all slightly different. Therefore companies like Motorola are releasing DSP chips which have a common *core* but have on-chip special purpose modules and interfaces. For example Motorola's DSP56156 has a 5616 core but with other modules, such as on-chip CODEC and PLL to tailor the chip for telecommunications applications. See also *DSP Processor*.

Correlation: If two signals are *correlated* then this means that they are in some sense similar. Depending on how similar they are, signals may be described as being *weakly correlated* or *strongly correlated*. If two signals, *x(k)* and *y(k)*, are ergodic then the correlation function, $r_{xy}(n)$ can be estimated as:

$$\hat{r}_{xy}(n) = \frac{1}{2M+1} \sum_{k=-M}^{M} x(k)y(n+k) \qquad \text{for large} M \tag{70}$$

Taking the discrete Fourier transform (DFT) of the autocorrelation function gives the cross spectral density. See also *Autocorrelation*.

Correlation Matrix: Assuming that a signal $x(k)$ is a wide sense stationary ergodic processes, a 3×3 *correlation matri*x can be formed by taking the expectation, $E\{.\}$, of the elements of the matrix formed by multiplying the signal vector, $\boldsymbol{x}(k) = [x(k) \;\; x(k-1) \;\; x(k-2)]$ by its transpose to produce the *correlation matrix*:

$$\begin{aligned} \boldsymbol{R} &= E[\boldsymbol{x}(k)\boldsymbol{x}^T(k)] = E\begin{bmatrix} x(k) \\ x(k-1) \\ x(k-2) \end{bmatrix} \begin{bmatrix} x(k) & x(k-1) & x(k-2) \end{bmatrix} \\ &= E\begin{bmatrix} x^2(k) & x(k)x(k-1) & x(k)x(k-2) \\ x(k)x(k-1) & x^2(k-1) & x(k-1)x(k-2) \\ x(k)x(k-2) & x(k-2)x(k-1) & x^2(k-2) \end{bmatrix} \\ &= \begin{bmatrix} r_0 & r_1 & r_2 \\ r_1 & r_0 & r_1 \\ r_2 & r_1 & r_0 \end{bmatrix} \end{aligned} \tag{71}$$

where $r_n = E[x(k)x(k-n)]$. The *correlation matrix*, $\boldsymbol{R}$ is Toeplitz symmetric and for a more general N point data vector the matrix will be N x N in dimension:

$$\boldsymbol{R} = \begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{N-1} \\ r_1 & r_0 & r_1 & \cdots & r_{N-2} \\ r_2 & r_1 & r_0 & \cdots & r_{N-3} \\ : & : & : & \cdots & : \\ r_{N-1} & r_{N-2} & r_{N-3} & \cdots & r_o \end{bmatrix} \tag{72}$$

The Toeplitz structure (i.e., constant diagonal entries) results from the fact that the diagonal entries all correspond to the same time lag estimate of the correlation, that is, $n+k-n = n$ is constant. To calculate r_n statistical averages should be used, or if the signal is ergodic then time averages can be used. See also *Adaptive Signal Processing, Cross Correlation Vector, Ergodic, Expected Value, Matrix, Matrix Structured - Toeplitz*, *Wide Sense Stationarity, Wiener-Hopf Equations.*

Correlation Vector: See *Cross Correlation Vector*.

CORTES Algorithm: Coordinate Reduction Time Encoding Scheme (CORTES) is an algorithm for the data compression of ECG signals. CORTES is based on the ATZEC and TP algorithms, using the AZTEC to discard clinically insignificant data in the isoelectric region, and applying the TP algorithm to clinically significant high frequency regions of the ECG data [48]. See also *AZTEC*, *Electrocardiogram, TP*.

Critical Bands: It is conjectured that a suitable model of the human auditory system is composed of a series of (constant fractional bandwidth) bandpass filters [30] which comprise *critical bands*. When trying to detect a signal of interest in broadband background noise the listener is thought to make use of a bandpass filter with a centre frequency close to that of the signal of interest. The perception to the listener is that the background noise is somewhat filtered out and only the components within the background noise that lie in the critical band remain. The threshold of hearing of the signal of interest is thus determined by the amount of noise passing through the filter. See also *Auditory Filters, Audiology, Audiometry, Fractional Bandwidth, Threshold of Hearing.*

Critical Distance: In a reverberant environment, the *critical distance* is defined as the separation between source and receiver that results in the acoustic energy of the reflected waveforms being equal to the acoustic energy in the direct path. A single number is often used to classify a given environment, although the specific acoustics of a given room may produce different *critical distances* for alternate source (or receiver) positions. Roughly, the critical distance characterizes how much reverberation exists in a given room. See also *Reverberation.*

Cross Compiler: This is a piece of software which allows a user to program in a high level language (such as 'C') and generate *cross compiled* code for the target DSP's assembly language. This code can in turn be assembled and the actual machine code program downloaded to the DSP processor. Although cross-compilers can make program writing much easier, they do not always produce efficient code (i.e. using minimal instructions) and hence it is often necessary to write in assembly language (or *hand code*) either the entire program or critical sections of the program (via in-line assembly commands in the higher level language program). Motorola produce a C *cross compiler* for the DSP56000 series, and Texas Instruments produce one for the TMS320 series of DSP processors.

Cross Correlation Vector: A 3 element *cross correlation vector*, $\boldsymbol{p}$, for a signal $d(k)$ and a signal $x(k)$ can be calculated from:

$$\boldsymbol{p} = E\{d(k)\boldsymbol{x}(k)\} = E\begin{bmatrix} d(k)x(k) \\ d(k)x(k-1) \\ d(k)x(k-2) \end{bmatrix} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix} \tag{73}$$

Hence for an *N* element vector:

$$\boldsymbol{p} = \begin{bmatrix} p_0 \\ p_1 \\ : \\ p_{N-1} \end{bmatrix} \tag{74}$$

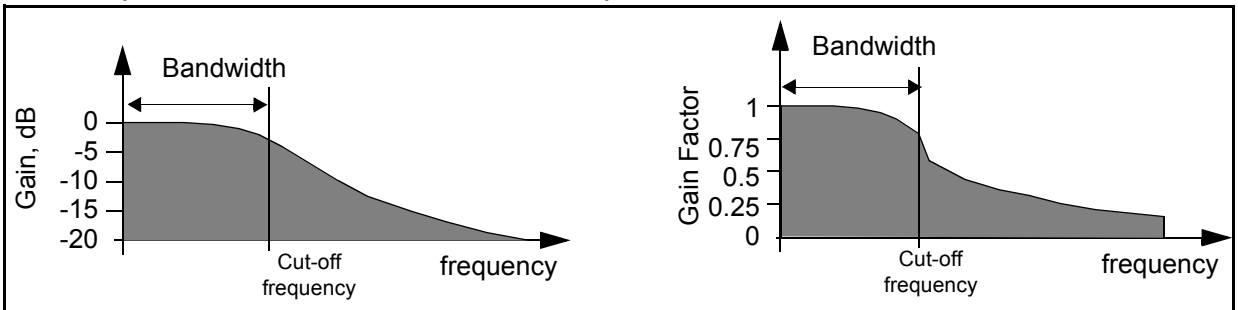
where $p_n = E\{d(k)x(k-n)\}$, and $E\{.\}$ is the expected value function. To calculate p_n statistical averages should be used, or if the signal is ergodic then time averages can be used. See also *Adaptive Signal Processing, Correlation Matrix, Ergodic, Matrix, Expected Value, Wide Sense Stationarity, Wiener-Hopf Equations*.

Cross Interleaved Reed Solomon Coding (CIRC): *CIRC* is an error correcting scheme which was adopted for use in compact discs (CD) systems [33]. *CIRC* is an interleaved combination of block (Reed-Solomon) and convolutional error correcting schemes. It is used to correct both burst errors and random bit errors. On a CD player errors can be caused by manufacturing defects, dust, scratches and so on. CIRC coding can be decoded to correct several thousand consecutive bit errors. It is safe to say that without the signal processing that goes into CD error correction and error concealment, the compact discs we see today would be substantially more expensive to produce and, subsequently, the CD players would not be nearly the ubiquitous appliance we see today. See also *Compact Disc.*

Cross-Talk: The interference of one channel upon another causing the signal from one channel to be detectable (usually at a reduced level) on another channel.

Cut-off Frequency: The *cut-off frequency* of a filter is the point at which the attenuation of the filter drops by 3dB. Although the term cut-off conjures up the image of a sharp attenuation, 3dB is equivalent to $20\log_{10}\sqrt{2}$, i.e. the filtered signal output has half of the power of the input signal, $10\log_{10}2$. For example the *cut-off frequency* of a low pass filter, is the frequency at which the filter

attenuation drops by 3dB when plotted on a log magnitude scale, and reduces by $\sqrt{2}$ on a linear scale. A bandpass filter will have two cut-off frequencies. See also *Attenuation, Decibels*



The cut-off frequency, or 3dB point of a filter. The left hand side illustrates the cut-off followed by the slow roll-off characteristic. The right hand side shows the same filter plotted as attenuation factor (linear scale, not decibel) against frequency. The cut off occurs when the attenuation is at $1/[\sqrt{2}]$

Cyberspace: The name given to the virtual dimension that the world wide network (internet) of connected computers gives rise to in the minds of people who spend a large amount of time "there". Without the DSP modems there would be no *cyberspace*! See also *Internet*.

Cyclic Redundancy Check (CRC): A *cyclic redundancy check* can be performed on digital data transmission systems whereby it is required at the receiver end to check the integrity of the data transmitted. This is most often used as an error detection scheme -- detected errors require retransmission. If both ends know the algebraic method of encoding the original data the raw data can be CRC coded at the transmission end, and then at the received end the cyclic (i.e., efficient) redundancy can be checked. This redundancy check highlights the fact that bit transmission errors have occurred. CRC techniques can be easily implemented using shift registers [40]. See also *Characteristic Polynomial, V-series Recommendations*.

Cyclostationary: If the autocorrelation function (or second order statistics) of a signal fluctuates *periodically* with time, then this signal is *cyclostationary*. See [75] for a tutorial article.

D

Damped Sinusoid: A common solution to linear system problems takes the form

$$e^{(a+jb)t} = e^{at}e^{jbt} = e^{at}[\cos(bt) + j\sin(bt)] \quad . \tag{75}$$

where the complex exponent gives rise to two separate components, an exponential decay term, e^{at} and a sinusoidal variation term $[\cos(bt) + j\sin(bt)]$. Common examples of systems that give rise to *damped sinusoidal* solutions are the suspension system in an automobile or the voltage in a passive electrical circuit that has energy storage elements (capacitors and inductors). Because many physical phenomena can be accurately described by coupled differential equations (for which *damped sinusoids* are common solutions), real world experiences of *damped sinusoids* are quite common.

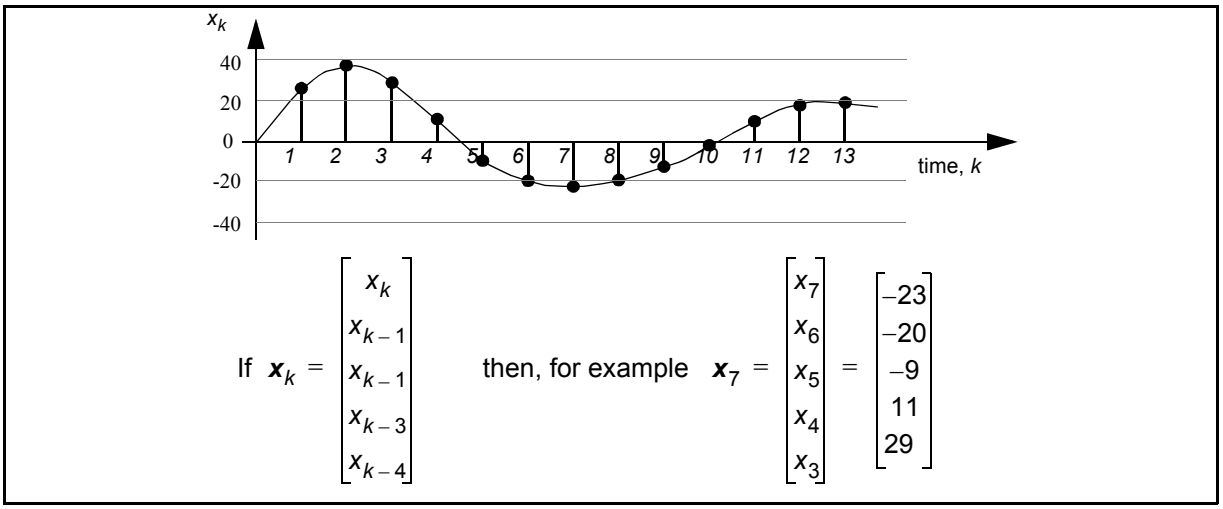
Data Acquisition: The general name given to the reading of data using an analog-to-digital converter (ADC) and storing the sampled data on some form of computer memory (e.g., a hard disk drive).

Data Bus: The *data bus* is a collection of wires on a DSP processor that is used to transmit actual data values between chips, or within the chip itself. See also *DSP Processor*.

Data Compression: See *Compression.*

Data Registers: Memory locations inside a DSP processor that can be used for temporary storage of data. The *data registers* are at least as long as the wordlength of the processor. Most DSP processors have a number of data registers. See also *DSP Processor.*

Data Vector: The most recent N data values of a particular signal, $x(k)$, can be conveniently represented as a vector, $\boldsymbol{x}_k$, where k denotes the most recent element in the vector. For example, if $N = 5$:



More generally any type of data stored or manipulated as a vector can reasonable be referred to as a *data vector*. See also *Vector, Vector Properties, Weight Vector*.

Data Windowing: See *Window.*

Daughter Module: Most DSP boards are designed to be hosted by an IBM PC. To provide input/output facilities or additional DSP processors some DSP boards (then called *motherboards*) have spaces for optional *daughter modules* to be inserted.

Decade: An *decade* refers the interval between two frequencies where one frequency is ten times other. Therefore as an example from 10Hz to 100Hz is a decade, and from 100Hz to 1000Hz is a decade and so on. See also *Logarithmic Frequency, Octave, Roll-off*.

Decibels (dB): The logarithmic unit of *decibels* is used to quantify power of any signal relative to a reference signal. A power signal dB measure is calculated as $10\log_{10}(P_1/P_0)$. In DSP since input signals are voltage, and Power = *(Voltage)*^{2} divided by *Resistance* we conventionally convert a voltage signal into its logarithmic value by calculating $20\log_{10}(V_1/V_0)$. *Decibels* are widely used to represent the attenuation or amplification of signals:

$$\text{Attentuation} = 10\log\left(\frac{P}{P_0}\right) = 20\log\left(\frac{V}{V_0}\right) \tag{76}$$

where P_o is the reference power, and V_0 is the reference voltage. dB's are used because they often provide a more convenient measure for working with signals (e.g., plotting power spectra) than do linear measures.

Often the symbol dB is followed by a letter that indicates how the *decibels* were computed. For example, dBm indicates a power measurement relative to a milliwatt, whereas dBW indicates power relative to a watt. In acoustics applications, dB can be measured relative to various perceptually relevant scales, such as A-weighting. In this case, noise levels are reported as dB(A) to indicate the relative weighting (A) selected for the measurement. See *Sound Pressure Level Weighting Curves, Decibels SPL.*

Decibels (dB) SPL: The *decibel* is universally used to measure acoustic power and sound pressure levels (SPL). The *decibel* rating for a particular sound is calculated relative to a reference power W_o:

$$10\log\left(\frac{W_1}{W_0}\right) \tag{77}$$

dB SPL is sound pressure measured relative to 20 μ-Pascals (2×10^{-5} Newtons/m^2). Acoustic power is proportional to pressure squared, so pressure based dB are computed via $20\log_{10}$ pressure ratios. Intensity (or power) based dB computations use $10\log_{10}$ intensity ratios. The sound level 0dB SPL is a low sound level that was selected to be around the absolute threshold of average human hearing for a pure 1000Hz sinusoid [30]. Normal speech has an SPL value of about 70dB SPL. The acoustic energy 200 feet from a jet aircraft at take-off about 125dB SPL, this is above the threshold of feeling (meaning you can *feel* the noise as well as hear it). See also *Sound Pressure Level.*

Decibels (dB) HL (3): Hearing Level (HL). See *Hearing Level, Audiogram.*

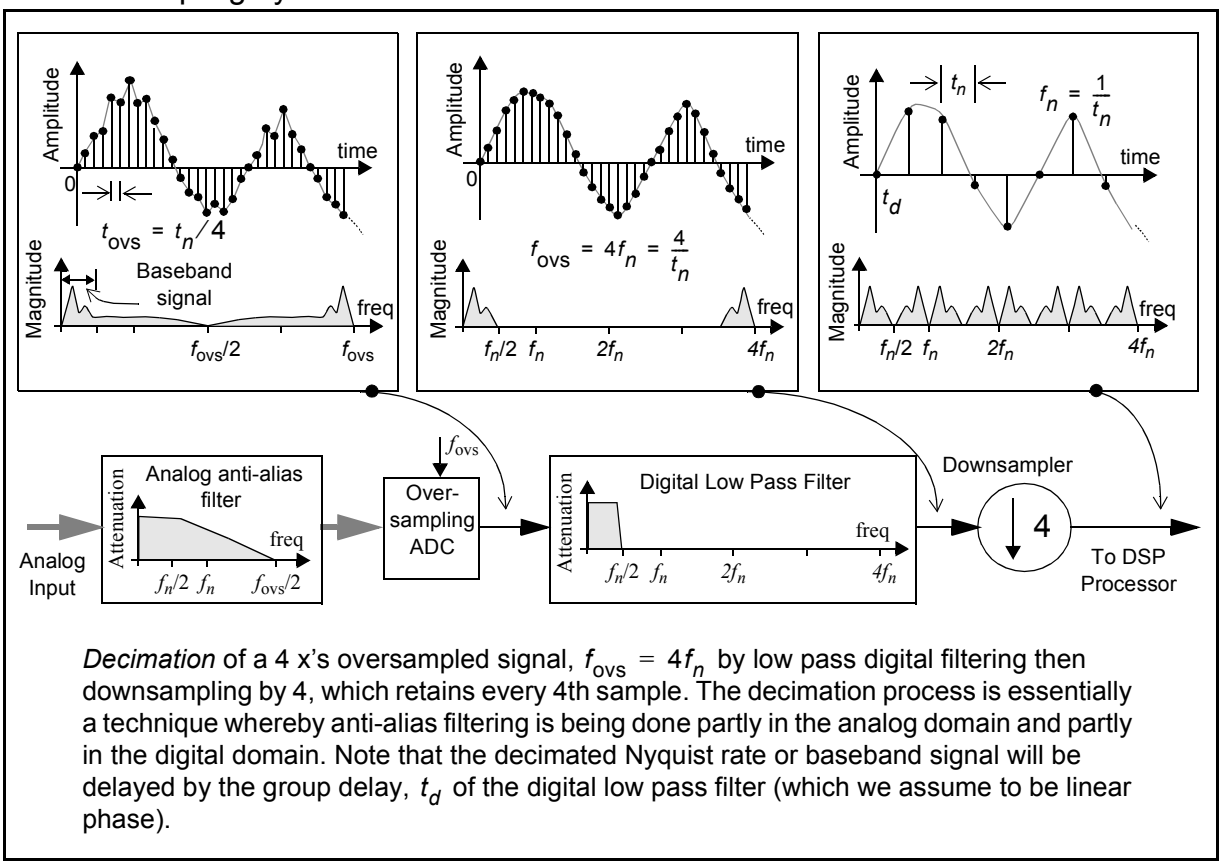
Decimation: Decimation is the process of reducing the sampling rate of a signal that has been oversampled. When a signal is bandlimited to a bandwidth that is a factor of 0.5 or less than half of the sampling frequency ($f_s/2$) then the sampling rate can be reduced without loss of information. Oversampling simply means that a signal has been sampled at a rate higher than dictated by the

Nyquist criteria. In DSP systems oversampling is usually done at integral multiples of the Nyquist rate, f_n, and usually by a power of two factor such as 4 x's, 8 x's or 64 x's.

For a discrete signal oversampled by a factor R, then the sampling frequency, f_s, is:

$$f_s \equiv f_{ovs} = Rf_n \tag{78}$$

For an R x's oversampled signal the only portion of interest is the baseband signal extending from 0 to $f_n/2$ Hz. Therefore *decimation* is required. The oversampled signal is first digitally low pass filtered to $f_n/2$ using a digital filter with a sharp cut-off. The resulting signal is therefore now bandlimited to $f_n/2$ and can be downsampled by retaining only every *R-th* sample. Decimation for a system oversampling by a factor of $R = 4$ can be illustrated as:



Decimation of a 4 x's oversampled signal, $f_{ovs} = 4f_n$ by low pass digital filtering then downsampling by 4, which retains every 4th sample. The decimation process is essentially a technique whereby anti-alias filtering is being done partly in the analog domain and partly in the digital domain. Note that the decimated Nyquist rate or baseband signal will be delayed by the group delay, t_d of the digital low pass filter (which we assume to be linear phase).

For the oversampling example above where $R = 4$, any frequencies that exist between $f_n/2$ Hz and $f_{ovs}/2 = 4f_n$ after the analog anti-alias filter can be removed with a *digital* low pass filter prior to downsampling by a factor of 4. Hence the complexity of the analogue low pass anti-alias filter has been reduced by effectively adding a digital low pass stage of anti-alias filtering.

So why not just oversample, but not decimate? To illustrate the requirement for decimation where possible, linear digital FIR filtering using an oversampled signal will require RN filter weights (corresponding to T secs) whereas the number of weights in the equivalent function Nyquist rate filter will only be N (also corresponding to T secs) Hence the oversampled DSP processing would require to perform R^2Nf_n multiply/adds per second, compared to the Nyquist rate DSP processing which requires Nf_n multiply/adds per second, a factor of R^2 more. This is clearly not very desirable and a considerable disadvantage of an oversampled system compared to a Nyquist rate system. Therefore this is why an oversampled signal is usually *decimated* to the Nyquist rate, first by digital low pass filtering, then by downsampling (retaining only every R-th sample).

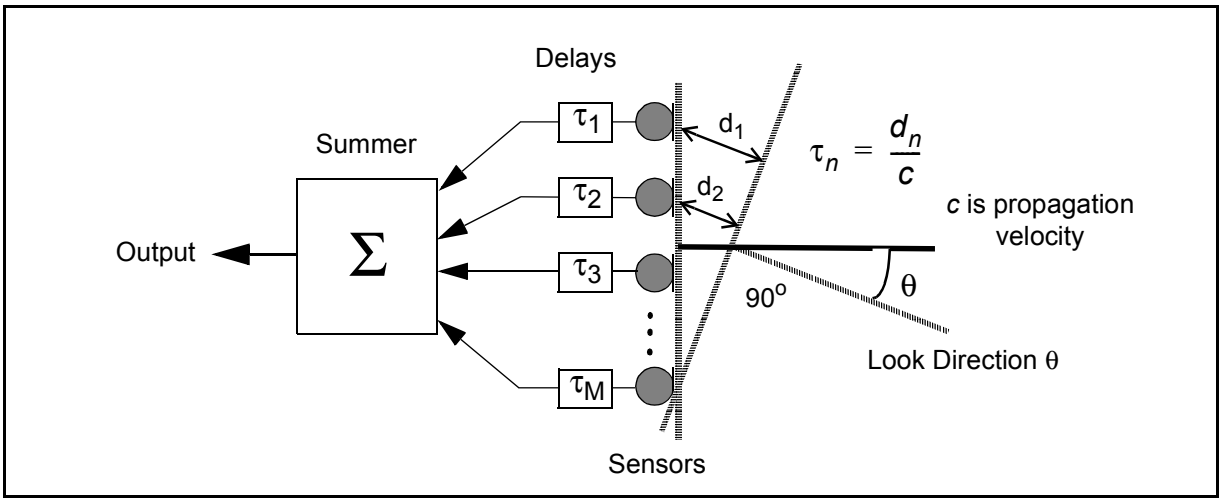
The word *decimation* originally comes from a procedure within the Roman armies, where for acts of cowardice the legionaires were lined up, and every 10th man was executed. Hence the prefix "dec" meaning ten.

See also *Anti-alias Filter, Downsampling, Oversampling, Upsampling, Interpolation, Sigma Delta.*

Decimation-in-Frequency (DIF): The DFT can be reformulated to give the FFT either as a DIT or a DIF algorithm. Since the input data and output data values of the FFT appear in bit-reversed order, *decimation-in-frequency* computation of the FFT provides the output frequency samples in bit-reversed order. See also *Bit Reverse Addressing, Discrete Fourier Transform, Fast Fourier Transform, Cooley-Tukey*.

Decimation-in-Time (DIT): The DFT can be reformulated to give the FFT either as a DIF or a DIT algorithm. Since the input data and output data values of the FFT appear in bit-reversed order, *decimation-in-time* computation of the FFT provides the output frequency samples in proper order when the input time samples are arranged in bit-reversed order. See also *Bit Reverse Addressing, Discrete Fourier Transform, Fast Fourier Transform, Cooley-Tukey*.

Delay and Sum Beamformer: A relatively simple beamformer in which the output from an array of sensors are subject to independent time delays and then summed together. The delays are typically selected to provide a look direction from which the desired signal will constructively interfere at the summer while signals from other directions are attenuated because they tend to destructively interfere. The delays are dictated by the geometry of the array of sensors and the speed of propagation of the wavefront. See also *Adaptive Beamformer, Beamformer, Broadside, Endfire.*

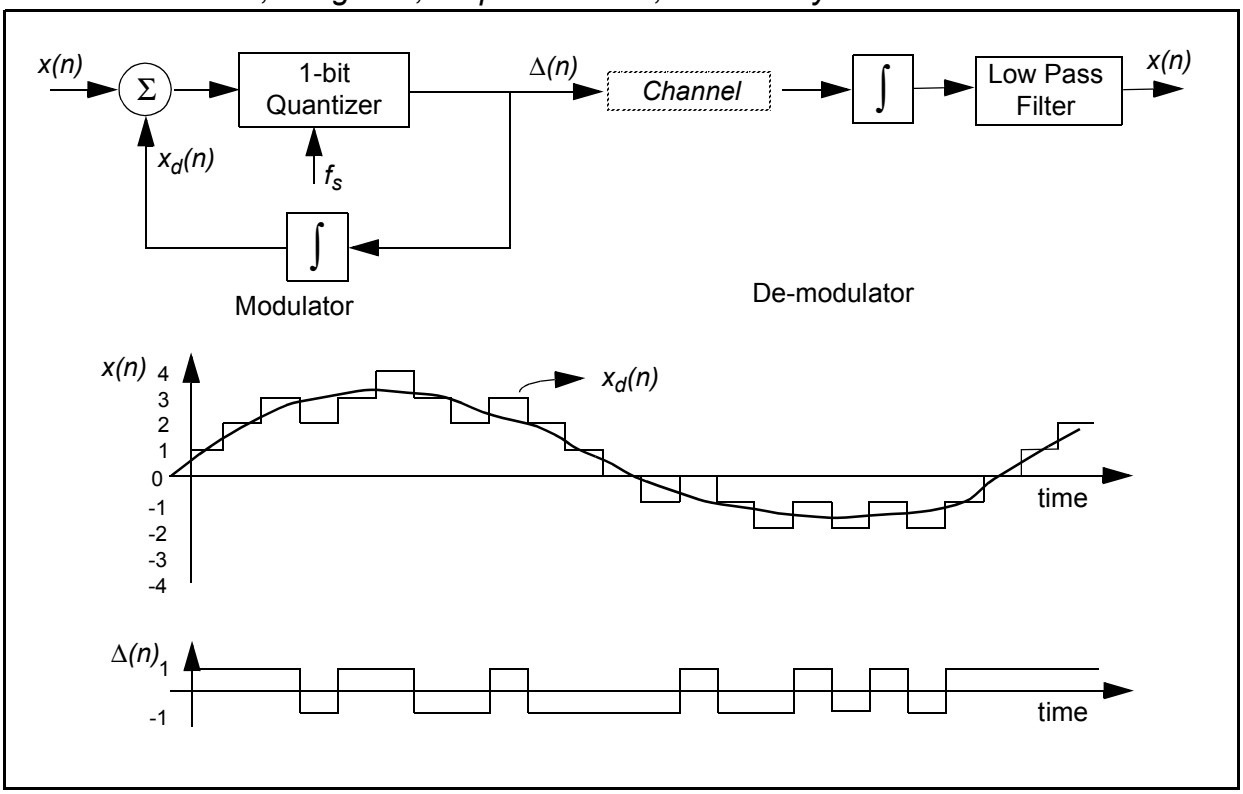


In a *delay-and-sum* beamformer, the output from each of the sensors in an array is delayed an appropriate amount (to time-align the desired signal) and then combined via a summation to generate the beamformed output. No amplitude weighting of the sensors is performed.

Delay LMS: See *Least Mean Squares Algorithm Variants.*

Delta Modulation: *Delta modulation* is a technique used to take a sampled signal, *x(n)*, and encode the magnitude change from the previous sample and transmit only the single bit difference (Δ) between adjacent samples [2]. If the signal has increased from the previous sample, then encode a 1, if it had decreased then encode as a -1. The received signal is then demodulated by taking successive delta samples and summing to reconstruct the original signal using an *integrator*. *Delta modulation* can reduce the number of bits per second to be transmitted down a channel,

compared to PCM. However when using a *delta modulator*, the sampling rate and step size must be carefully chosen or slope overload and/or granularity problems may occur. See also *Adaptive Differential Pulse Code Modulation,Continuously Variable Slope Delta Modulation, Differential Pulse Code Modulation, Integrator, Slope Overload, Granularity Effects.*.



Delta-Sigma: Synonymous term with Sigma Delta. See *Sigma-Delta*.

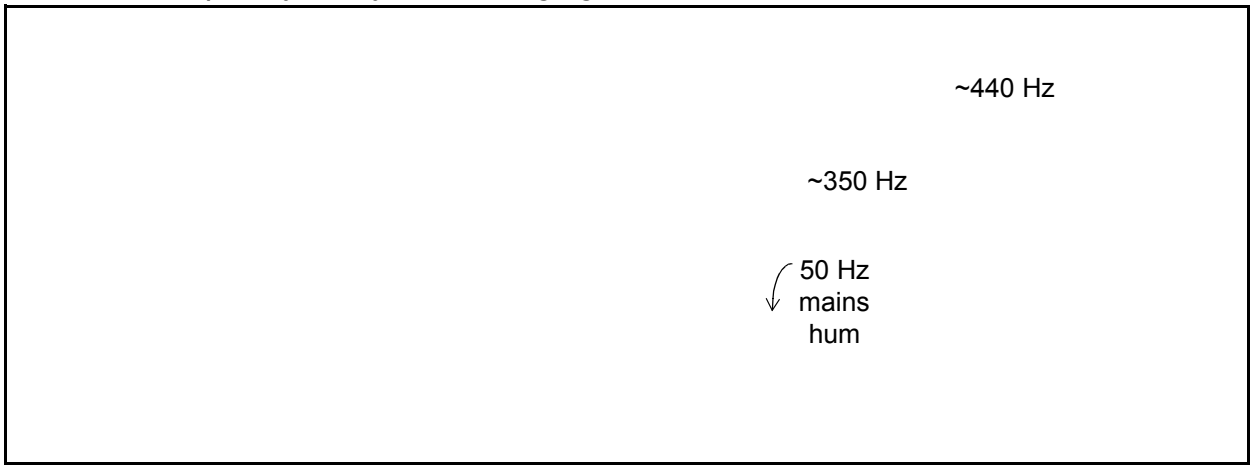
Descrambler: See *Scrambler/Descrambler*.

Destructive Interference: The addition of two waveforms with nearly opposite phase. *Destructive interference* is exploited to cancel unwanted noise, vibrations, and interference in physical and electrical systems. *Destructive interference* is also responsible for energy nulls in diffraction patterns. See also *Diffraction*, *Constructive Interference, Beamforming.*

Determinant: See *Matrix Properties - Determinant.*

Diagonal Matrix: See *Matrix Structured - Diagonal.*

Dial Tone: Tones at 350 Hz and 440 Hz make up the dialing tone for telephone systems. See also *Dual Tone Multifrequency, Busy Tone, Ringing Tone*.



Dichotic: A situation where the aural stimulation reaching both ears is not the same. For example, setting up a demonstration of binaural beats is a *dichotic* stimulus. The human ear essentially provides *dichotic* hearing whereby it is possible for the auditory mechanism to process the differing information arriving at both ears and subsequently localize the source. See also *Audiometry, Binaural Unmasking, Binaural Beats,Diotic, Lateralization, .*

Difference Limen (DL): The smallest noticeable difference between two audio stimuli, or the Just Noticeable Difference (JND) between these stimuli. Determination of *DL*'s usually requires that subjects be given a discrimination task. Typically, *DL*'s (or JND's) are computed for two signals that are identical in all respects save the parameter being tested for a *DL*. For example, if the *DL* is desired for sound intensity discrimination, two stimuli differing only in intensity would be presented to the subject under test. These stimuli could be tones at a given frequency that are presented for a fixed period. It is interesting to note that the *DL* for sound intensity (measured in dB) is generally found to be constant over a very wide range (this is known as Weber's law).

To have meaning a *DL* must be specified along with the set up and conditions used to establish the value. For example stating that the frequency *DL* for the human ear is 1Hz between the frequencies of 1- 4 kHz requires that sound pressure levels, stimuli duration, and stimuli decomposition are clearly stated as varying these parameters will cause variation in the measured frequency *DL*. See also *Audiology, Audiometry, Frequency Range of Hearing, Threshold of Hearing.*

Differentiation: See *Differentiator*.

Differential Phase Shift Keying (DPSK): A type of modulation in which the information bits are encoded in the *change* of the relative phase from one symbol to the next. *DPSK* is useful for communicating over time varying channels. *DPSK* also removes the need for absolute phase synchronization, since the phase information is encoded in a relative way. See also *Phase Shift Keying.*

Differentiator: A (linear) device that will produce an output that is the derivative of the input. In digital signal processing terms a *differentiator* is quite straightforward. The output of a *differentiator*, y(t), will be the rate of change of the signal curve, x(t), at time t. For sampled digital signals the input will be constant for one sampling period, and therefore to differentiate the signal the previous sample value is subtracted from the current value and divided by the sampling period. If the sampling period is normalized to one, then a signal is differentiated in the discrete domain by

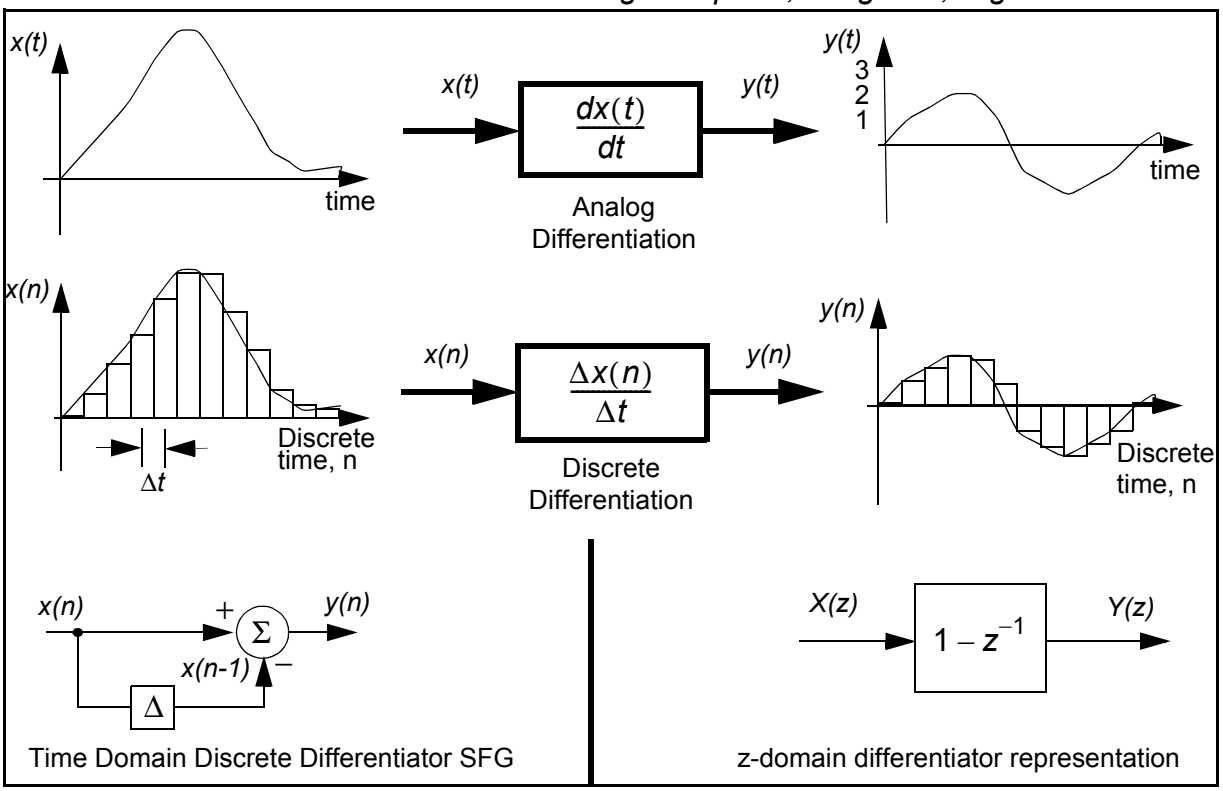
subtracting consecutive input samples. A *differentiator* is implemented using a digital delay element, and a summing element to calculate:

$$y(n) = x(n) - x(n-1) \tag{79}$$

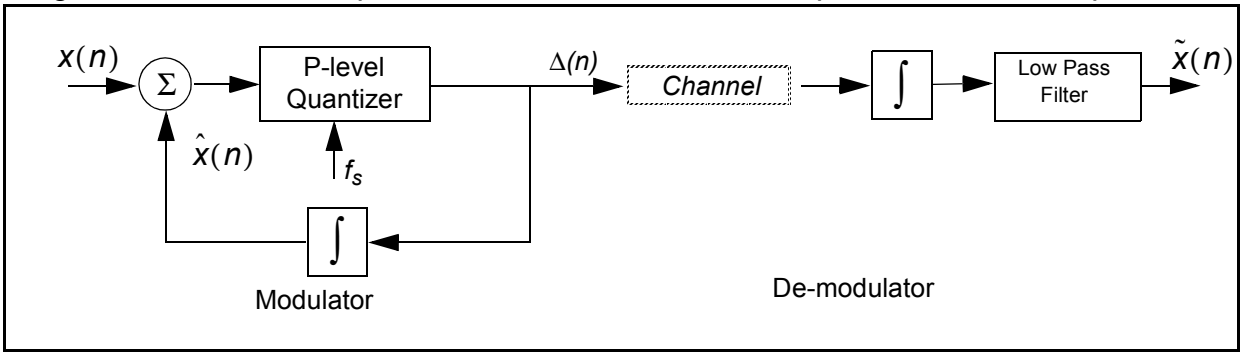
In the z-domain the transfer function of a *differentiator* is:

$$\begin{aligned} Y(z) &= X(z) - z^{-1}X(z) \\ \Rightarrow \quad \frac{Y(z)}{X(z)} &= 1 - z^{-1} \end{aligned} \tag{80}$$

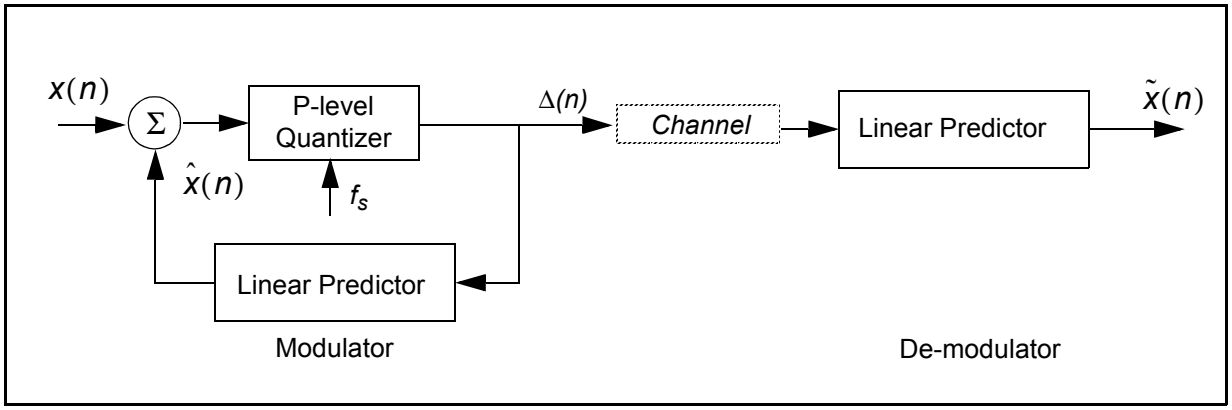
When viewed in the frequency domain a *differentiator* has the characteristics of a high pass filter. Thus differentiating a signal with additive noise tends to emphasize or enhance the high frequency components of the additive noise. See also *Analog Computer, Integrator, High Pass Filter*.



Differential Pulse Code Modulation (DPCM): *DPCM* is an extension of delta modulation that makes use of redundancy in analog signals to quantize the difference between a discrete input signal and a predicted value to one of *P* values [2]. (Note a delta modulator has only one level ±1). The integrator shown below performs a summation of all input values as the predictor. More

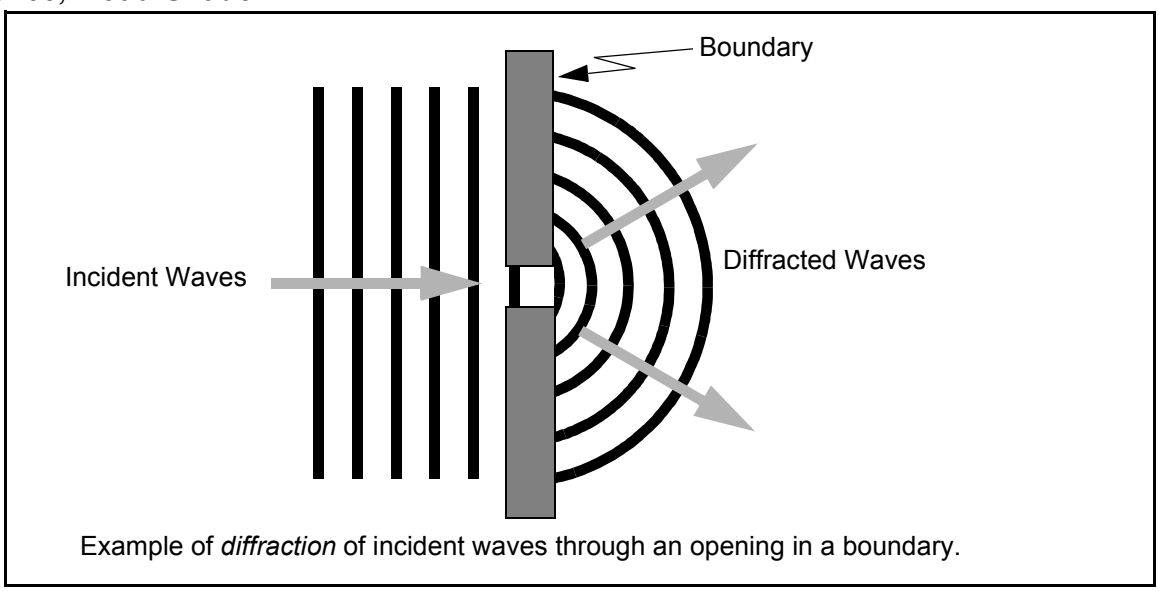


complex *DPCM* systems require a predictor filter in place of the simple integrator. Note that the



predictor at the modulator end uses the same quantized error values as inputs that are available to the predictor at the demodulator end. If the unquantized error values were used at the modulator end then there would be an accumulated error between demodulator output and the modulator input with a strictly increasing variance. This does not happen in the above configuration. See also *Adaptive Differential Pulse Code Modulation (ADPCM), Delta Modulation, Continuously Variable Slope Delta Modulation (CVSD), Slope Overload, Granularity*.

Diffraction: *Diffraction* is the bending of waves around an object via wave propagation of incident and reflected waves impinging on the object. See also *Constructive Interference, Destructive Interference, Head Shadow.*



Example of *diffraction* of incident waves through an opening in a boundary.

Digital: Represented as a discrete countable quantity. When an analog voltage is passed through an ADC the output is a digitized and sampled version of the input. Note that digitization implies quantization.

Digital Audio: Any aspect of audio reproduction or recording that uses a digital representation of analogue acoustic signals is often referred to generically as *digital audio* [33], [34], [37]. Over the last 10-20 years *digital audio* has evolved into three distinguishable groups of application dependent quality:

1. **Telephone Speech 300 - 3400Hz:** Typically speech down a telephone line is carried over a channel with a bandwidth extending from around 300Hz to 3400Hz. This bandwidth is adequate for good coherent and intelligible conversation. Music is coherent but unattractive. Clearly intelligible speech can be obtained by

sampling at 8kHz with 8 bit PCM samples, corresponding to an uncompressed bit rate of 64kbits/s.

2. **Wideband Speech: 50 - 7000Hz:** For applications such as teleconferencing prolonged conversation requires a speech quality that has more naturalness and presence. This is accomplished by retaining low and high frequency components of speech compared to a telephone channel. Music with the same bandwidth will have almost AM radio quality. Good quality speech can be obtained by sampling at 16kHz with 12 bit PCM samples, corresponding to a bit rate of 192kbits/s.
3. **High Fidelity Audio: 20 - 20000Hz:** For high fidelity music reproduction audio the reproduced sound should be of comparable quality to the original sound. Wideband audio is sampled at one of the standard frequencies of 32 kHz, 44.1 kHz, or 48 kHz using 16 bit PCM. A stereo compact disc (44.1kHz, 16 bits) has a data rate of 1.4112 Mbits/s.

Generally, when one refers to digital audio applications involving speech materials only (e.g., speech coding) the term speech is directly included in the descriptive term. Consequently, digital audio has come to connote high fidelity audio, with speech applications more precisely defined.

The table below summarizes the key parameters for a few well known *digital audio* applications. Note that to conserve bandwidth and storage requirements DSP enabled compression techniques are applied in a few of these applications.

Technology	Example Application	Sampling Rate (kHz)	Com-pression	Single Channel Bit Rate (kbits/s)
Digital Audio Tape (DAT)	Professional recording	48	No	768
Compact Disc (CD)	Consumer audio	44.1	No	705.6
Digital Compact Cassette (DCC)	Consumer audio	32, 44.1, 48	Yes	192
MiniDisc (MD)	Consumer audio	44.1	Yes	146
Dolby AC-2	Cinema sound	48	Yes	128
MUSICAM (ISO Layer II)	Consumer broadcasting	32, 44.1, 48	Yes	16 - 192
NICAM	TV audio	32	Yes	338
PCM A/μ-law (G711)	Telephone	8	Yes	64
ADPCM (G721)	Telephone	8	Yes	16,24,32,40
LD-CELP (G728)	Telephone	8	Yes	16
RPE-LTP (GSM)	Telephone	8	Yes	13.3
Subband ADPCM (G722)	Teleconferencing	16	Yes	64

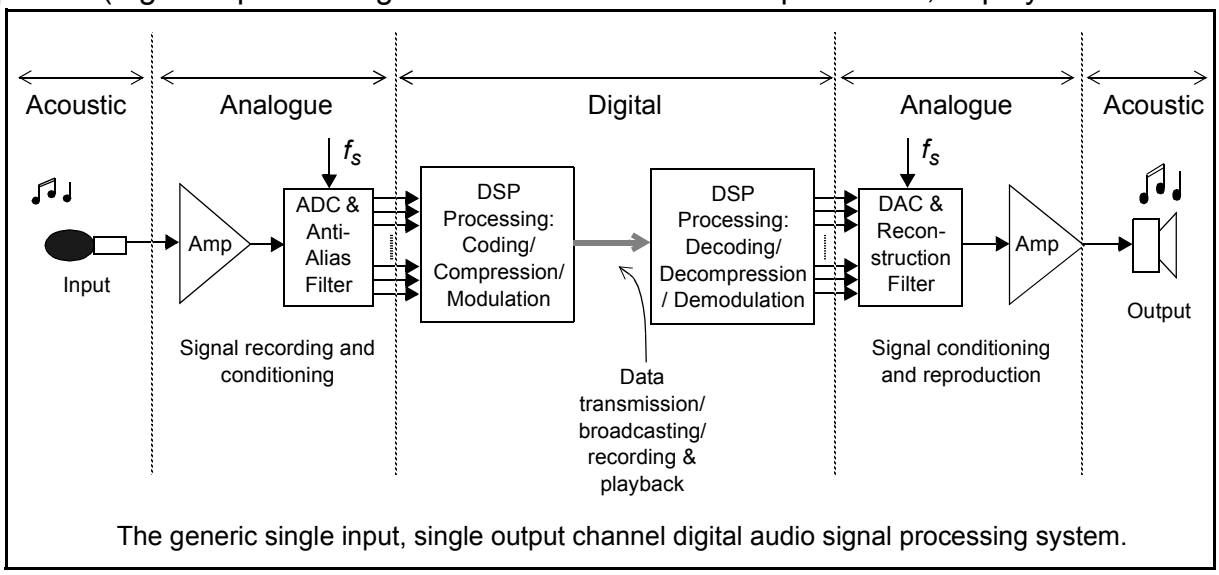
Digital Audio Systems

Although the *digital audio* market is undoubtedly very mature, the power of DSP systems is stimulating research and development in a number of areas:

1. Improved compression strategies based on perceptual and predictive coding; compression ratios of up to 20:1 for hifidelity audio may eventually be achievable.
2. The provision of surround sound using multichannel systems to allow cinema and “living room” audiences to experience 3-D sound.
3. DSP effects processing: remastering, *de-scratching* recordings, sound effects, soundfield simulation etc.
4. Noise reduction systems such as adaptive noise controllers, echo cancellers, acoustic echo cancellers, equalization systems.

5. *Super-fidelity* systems sampling at 96kHz to provide *ultrasound* [154] (above 20kHz and which is perhaps more tactile than audible), and systems to faithfully reproduce infrasound [138] (below 20Hz and which is most definitely tactile and in some cases rather dangerous!)

Real-time *digital audio* systems are one of three types: (1) input/output system (e.g. telephone/ teleconferencing system); (2) output only (e.g. CD player); or (3) input only (e.g. DAT professional recording). The figure below shows the key elements of a single channel input/output *digital audio* system. The input signal from a microphone is signal conditioned/amplified as appropriate to the input/output characteristic of the analogue to digital converter (ADC) at a sampling rate of f_s Hz. Prior to being input to the ADC stage the analogue signal is low pass filtered to remove all frequencies above $f_s/2$ by the analogue anti-alias filter. The output from ADC is then a stream of binary numbers, which are then compressed, coded and modulated for transmission, broadcasting or recording via/to a suitable medium (e.g. FM radio broadcast, telephone call or CD mastering). When a *digital audio* signal is received or read it is a stream of binary numbers which are demodulated and decoded/decompressed with DSP processing into a sampled data PCM format for input to a digital to analogue converter (DAC) which outputs to an analogue low pass reconstruction filter stage (also cutting off at $f_s/2$ prior to being amplified and output to a loudspeaker (e.g. reception of digital audio FM radio or a telephone call, or playback of a CD).



The generic single input, single output channel digital audio signal processing system.

See also *Compact Disc, Data Compression, Digital Audio Tape, Digital Compact Cassette, MiniDisc, Speech Coding*.

Digital Audio Broadcasting (DAB): The transmission of electromagnetic carriers modulated by digital signals. DAB will permit the transmission of high fidelity audio and is more immune to noise and distortion than conventional techniques. Repeater transmitters can receive a DAB signal, clean the signal and retransmit a noise free version. Currently there is a large body of interest in developed DAB consumer systems using a combination of satellite, terrestrial and cable transmission. For terrestrial DAB however there is currently no large bandwidth specifically allocated for DAB, and therefore FM radio station owners may be required to volunteer their bands for *digital audio broadcasting*. See also *Compression,Standards*.

Digital Audio Tape (DAT): An audio format introduced in the late 1980s to compete with compact disc. DAT samples at 48kHz and used 16 bit data with stereo channels. Although DAT was a commercial failure for the consumer market it has been adopted as a professional studio recording

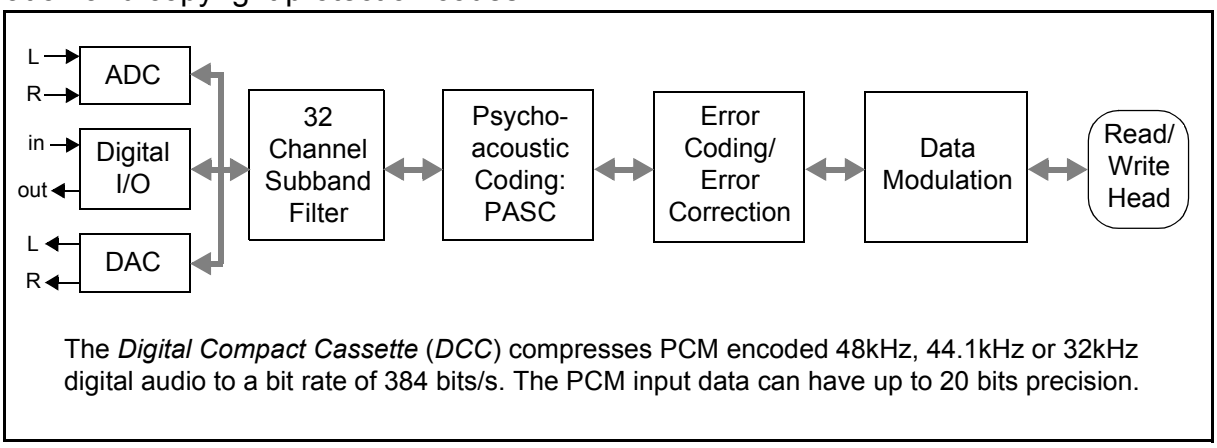
medium. A very similar format of 8mm digital tape is also quite commonly used for data storage. See also *Digital Compact Cassette, MiniDisc.*

Digital Communications: The process of transmitting and receiving messages (information) by sending and decoding one of a *finite* number of symbols during a sequence of symbol periods. One primary requirement of a digital communication system is that the information *must* be represented in a digital (or discrete) format. See also *Message,Symbol, Symbol Period.*

Digital Compact Cassette (DCC): *DCC* was introduced by Philips in the early 1990s as a combination of the physical format of the popular compact cassette, and featuring new digital audio signal processing and magnetic head technology [83], [52], [150]. Because of physical constraints *DCC* uses psychoacoustic data compression techniques to increase the amount of data that can be stored on a tape. The *DCC* mechanism allows it to play both (analog) compact cassette tapes and *DCC* tapes. The tape speed is 4.75cm/s for both types of tapes and a carefully designed thin film head is used to achieve both digital playback and analog playback. The actual tape quality is similar to that used for video tapes. *DCC* is a competing format to Sony's MiniDisc which also uses psychoacoustic data compression techniques.

If normal stereo 16 bit, 48kHz (1.536 Mbits/sec) PCM digital recording were done on a *DCC* tape, only about 20 minutes of music could be stored due to the physical restrictions of the tape. Therefore to allow more than an hour of music on a single tape data compression is required. *DCC* uses precision adaptive subband coding (PASC) to compress the audio by a factor of 4:1 to a data rate of 384 Mbits/s (192 Mbits/s per channel) thus allowing more than an hour of music to be stored. PASC is based on psychoacoustic compression principles and is similar to ISO/MPEG layer 1 standard. The input to a PASC encoder can be PCM data of up to 20 bits resolution at sampling rates of 48kHz, 44.1kHz or 32kHz. The quality of music from a PASC encoded DCC is arguably as good as a CD, and in fact for some parameters such as dynamic range a prerecorded DCC tape can have improved performance over a CD (see *Precision Adaptive Subband Coding*).

Eight to ten modulation and cross interleaved Reed-Solomon coding (CIRC) is used for the *DCC* tape channel coding and error correction. In addition to the audio tracks *DCC* features an auxiliary channel capable of storing 6.75kbits/sec and which can be used for storing timing, textual information and copyright protection codes.

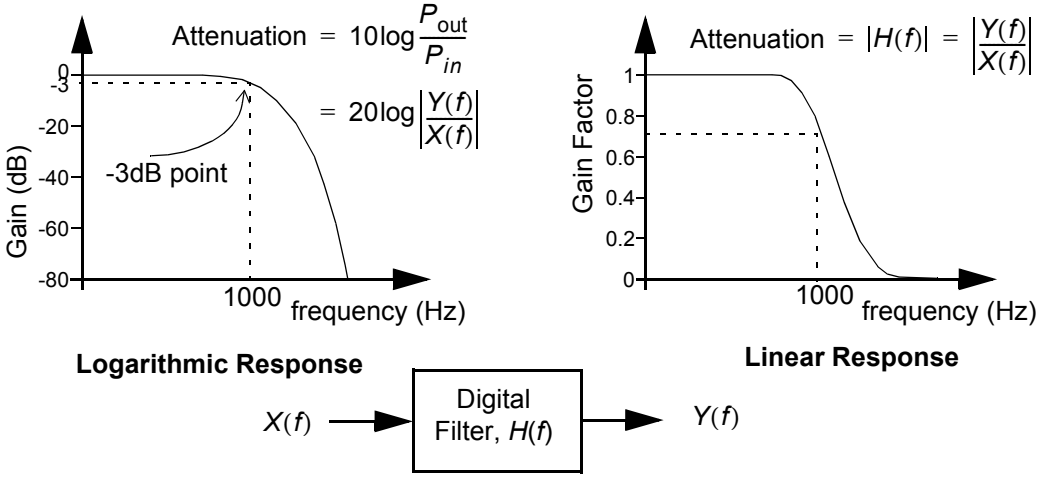


The *Digital Compact Cassette* (*DCC*) compresses PCM encoded 48kHz, 44.1kHz or 32kHz digital audio to a bit rate of 384 bits/s. The PCM input data can have up to 20 bits precision.

In terms of DSP algorithms the *DCC* also uses an IIR digital filter for equalization of the thin film magnetic head frequency response, and a 12 weight FIR filter to compensate for the high frequency roll-off of the magnetic channel. See also *Compact Disc, Digital Audio, Digital Audio Tape (DAT), MiniDisc, Precision Adaptive Subband Coding (PASC), Psychoacoustics*.

Digital European Cordless Telephone (DECT): The DECT is a telephone whereby a wireless radio connection at 1.9GHz communicates with a base station and is normally connected to the public switched telephone network. One or more handsets can communicate with each other or the outside world.

Digital Filter: A DSP system that will filter a digital input (i.e., selectively discriminate signals in different frequency bands) according to some pre-designed criteria is called a *digital filter*. In some situations *digital filters* are used to modify phase only [10], [7], [21], [31], [29]. A *digital filter*'s characteristics are usually viewed via their frequency response and for some applications their phase response (discussed in *Finite Impulse Response Filter*, and *Infinite Impulse Response Filter*). For the frequency response, the filter attenuation or gain characteristic can either be specified on a linear gain scale, or more commonly a logarithmic gain scale:

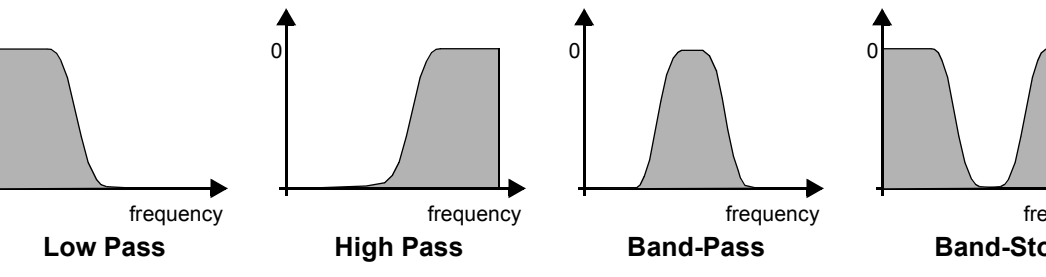


The above *digital filter* is a low pass filter cutting off at 1000Hz. Both the linear and logarithmic magnitude responses of the transfer function, $H(f) = Y(f)/X(f)$ are shown. The cut-off frequency of a filter is usually denoted as the "3dB frequency", i.e. at f_{3dB} = 1000 Hz, the filter attenuates the power of a sinusoidal component signal at this frequency by 0.5, i.e.

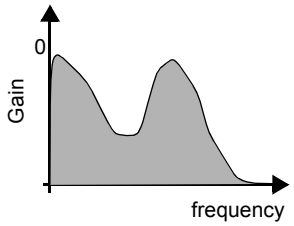
$$10\log\frac{P_{out}}{P_{in}}\bigg|_{f3dB} = 20\log\left|\frac{Y(f_{3dB})}{X(f_{3dB})}\right| = 10\log 0.5 = 20\log 0.707\ldots = -3\ \text{dB}$$

The power of the output signal relative to the input signal at f_{3dB} is therefore 0.5, and the signal amplitude is attenuated by $1/\sqrt{2} = 0.707\ldots$. For a low pass filter signals with a frequency higher than f_{3dB} are attenuated by more than 3dB.

Digital filters are usually designed as either low pass, high pass, band-pass or band-stop:



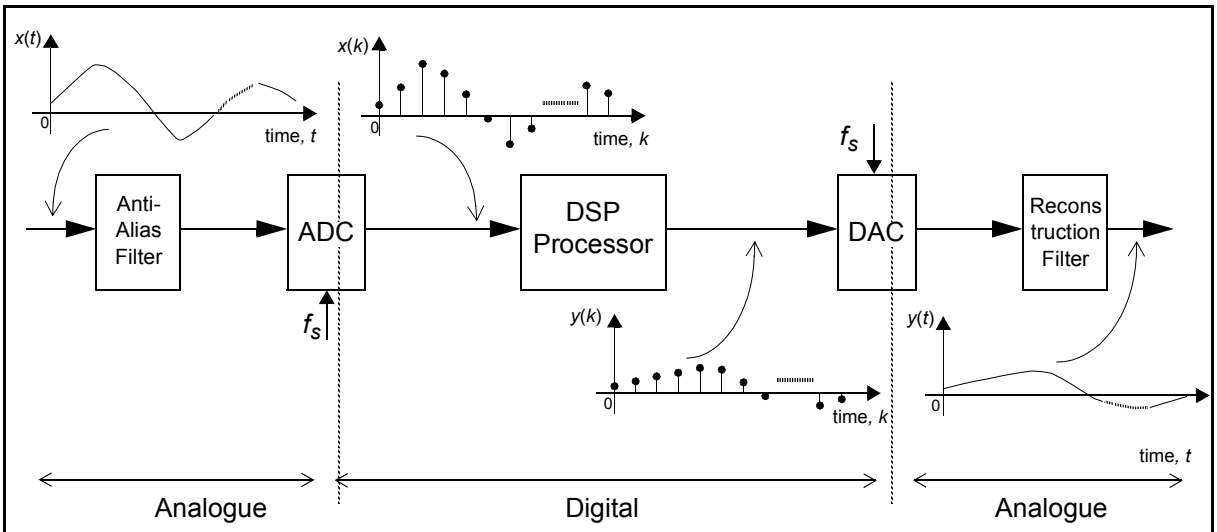
A number of filter design packages will give the user the facility to design a filter for an arbitrary frequency response by “sketching” graphically:



User Defined Frequency Response

There are two types of linear digital filters, FIR (finite impulse response filter) and IIR (infinite impulse response filter). An *FIR* filter is a digital filter that performs a moving, weighted average on a discrete input signal, $x(n)$, to produce an output signal. (For a more intuitive discussion of FIR filtering operation see entry for *Finite Impulse Response Filter*).

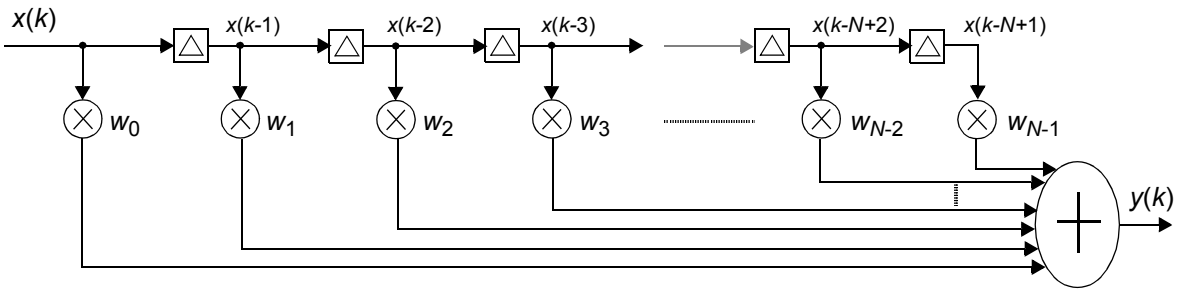
The arithmetic computation required by the digital filter is of course performed on a DSP processor or equivalent:



The digital filter equations are implemented on the DSP Processor which processes the time sampled data signal to produce a time sampled output data signal.

The actual frequency and phase response of the filter is found by taking the discrete frequency transform (DFT) of the weight values of w_0 to w_{N-1}.

An FIR digital filter is usually represented in a signal flow graph or with a summation (convolution) equation:



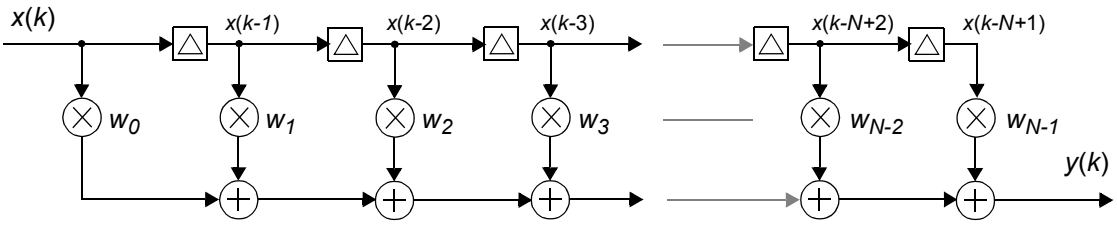
$$y(k) = w_0x(k) + w_1x(k-1) + w_2x(k-2) + w_3x(k-3) + ... + w_{N-1}x(k-N+1)$$

$$= \sum_{n=0}^{N-1} w_n x(k-n) = \boldsymbol{w}^T\boldsymbol{x}_k$$

where $\boldsymbol{w} = \left[w_0\ w_1\ w_2\ \ldots\ w_{N-1}\right]$ and $\boldsymbol{x}_k = \left[x(k)\ x(k-1)\ x(k-2)\ :\ x(k-N+1)\right]$

The signal flow graph and the output equation for an FIR *digital filter*. The filter output $y(k)$ can be expressed as a summation equation, a difference equation or using vector notation.

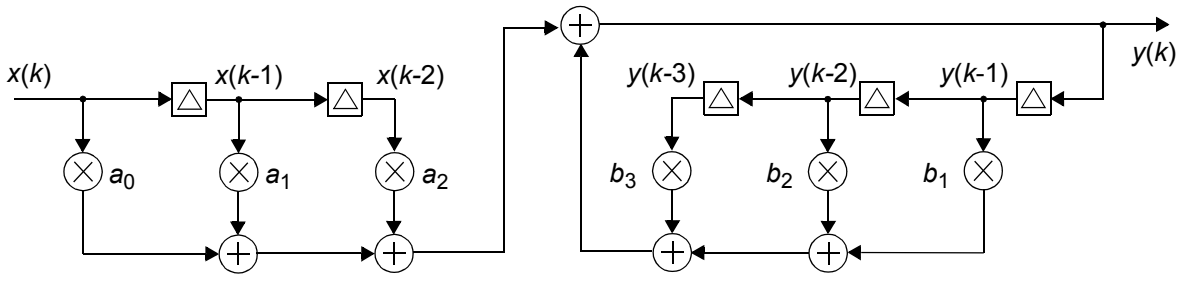
The signal flow graph can be drawn in a more modular fashion by splitting the *N* element summer into a series of two element summers:



The signal flow graph for an FIR filter is often modularized in order that the large *N* element summer is broken down into a series of *N*-1 two element summing nodes. The operation, of course, of this filter is identical to the above.

An IIR digital utilizes feedback (or recursion) in order to achieve a longer impulse response and therefore the possible advantage of a filter with a sharper cut off frequency (i.e., smaller transition bandwidth - see below) but with fewer weights than an FIR digital filter with an analogous frequency response. (For a more intuitive discussion on the operation of an IIR filter see entry for *Infinite Impulse Response Filter*.) The attraction of few weights is that the filter is cheaper to implement (in

terms of power consumption, DSP cycles and/or cost of DSP hardware). The signal flow graph and output equation for an IIR filter is:



$$y(k) = \sum_{n=0}^{2} a_n x(k-n) + \sum_{n=1}^{3} b_n y(k-n)$$

$$= a_0 x(k) + a_1 x(k-1) + a_2 x(k-2) + b_1 y(k-1) + b_2 y(k-2) + b_3 y(k-3)$$

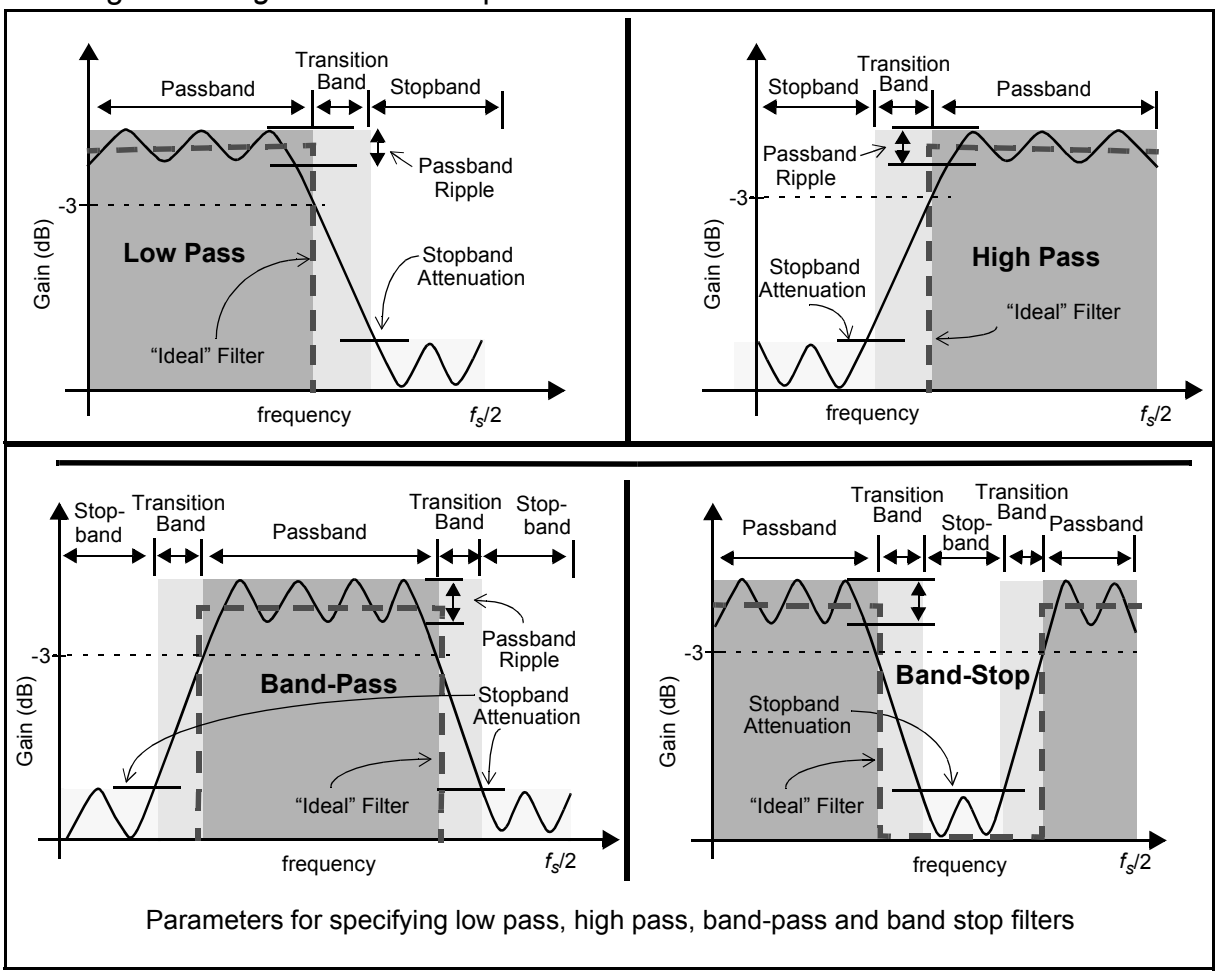
$$= \boldsymbol{a}^T \boldsymbol{x}_k + \boldsymbol{b}^T \boldsymbol{y}_{k-1} = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} x(k) \\ x(k-1) \\ x(k-2) \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} y(k-1) \\ y(k-2) \\ y(k-3)) \end{bmatrix}$$

A signal flow graph and equation for a 2 zero, 3 pole IIR digital filter. The filter output $y(k)$ can be expressed as a summation equation, a difference equation or using vector notation.

Design algorithms to find suitable weights for digital FIR filters are incorporated into many DSP software packages and typically allow the user to specify the parameters of:

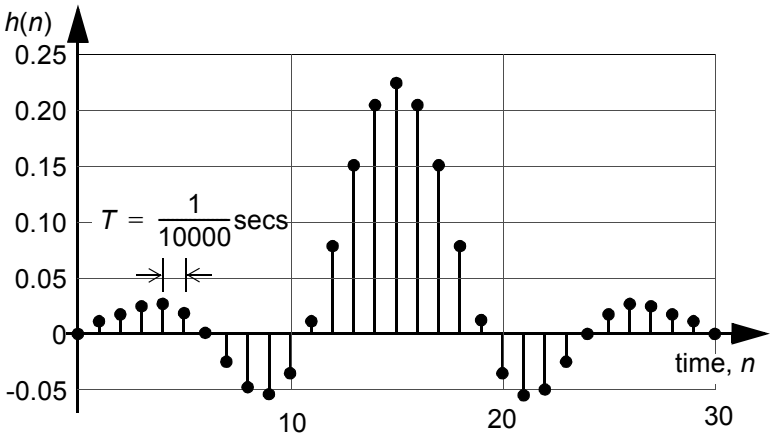
- Sampling frequency;
- Passband;
- Transition band;
- Stopband;
- Passband ripple;
- Stopband attenuation;
- No. of weights in the filter.

These parameters allow variations from the ideal (brick wall) filter, with the trade-offs being made by the design engineer. In general, the less stringent the bounds on the various parameters, then the fewer weights the *digital filter* will require:



Parameters for specifying low pass, high pass, band-pass and band stop filters

After the filter weights are produced by DSP filter design software the impulse response of the digital filter can be plotted, i.e. the filter weights shown against time:

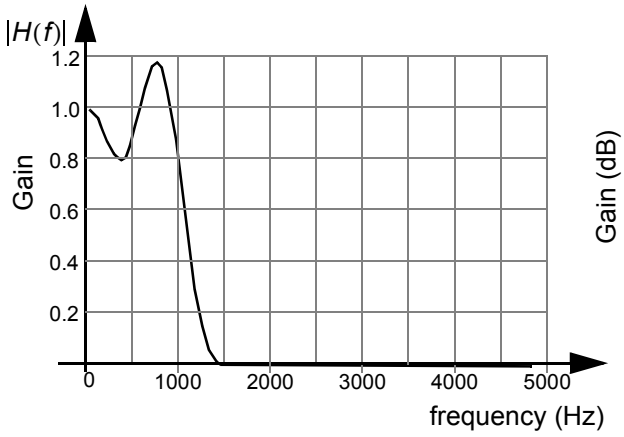


$w_0 = w_{30}$ = 0.00378...
$w_1 = w_{29}$ = 0.00977...
$w_2 = w_{28}$ = 0.01809...
$w_3 = w_{27}$ = 0.02544...
$w_4 = w_{26}$ = 0.027154...
$w_5 = w_{25}$ = 0.019008...
$w_6 = w_{24}$ = 0.00003...
$w_7 = w_{23}$ = -0.02538...
$w_8 = w_{22}$ = -0.04748...
$w_9 = w_{21}$ = -0.05394...
$w_{10} = w_{20}$ = -0.03487...
$w_{11} = w_{19}$ = 0.01214...
$w_{12} = w_{18}$ = 0.07926...
$w_{13} = w_{17}$ = 0.14972...
$w_{14} = w_{16}$ = 0.20316...
w_{15} = 0.22319...
(Truncated to 5 decimal places)

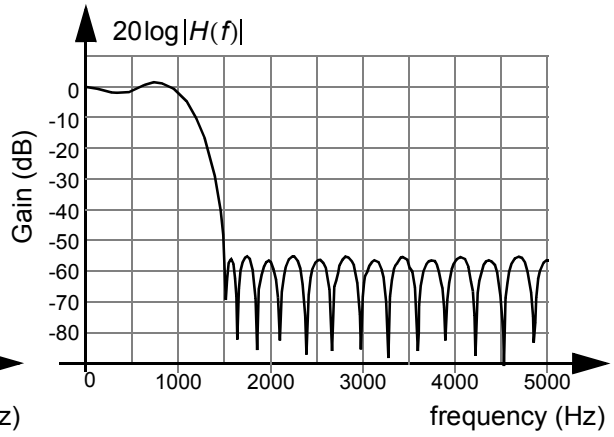
DESIGN 1: Low Pass FIR Filter Impulse Response

The impulse response $h(n) = w_n$ of the low pass filter specified in the above SystemView dialog boxes: cut-off frequency 1000 Hz; passband gain 0dB; stopband attenuation 60dB; transition band 500 Hz; passband ripple 5dB and sampling at f_s = 10000 Hz. The filter is linear phase and has 31 weights and therefore an impulse response of duration 31/10000 seconds. For this particular filter the weights are represented with floating point real numbers. Note that the filter was designed with 0dB in the passband. As a quick check the sum of all of the coefficients is approximately 1, meaning that if a 0 Hz (DC) signal was input, the output is not amplified or attenuated, i.e. gain = 1 or 0 dB.

From the impulse response the DFT (or FFT) can be used to produce the filter magnitude frequency response and the actual filter characteristics can be compared with the original desired specification:



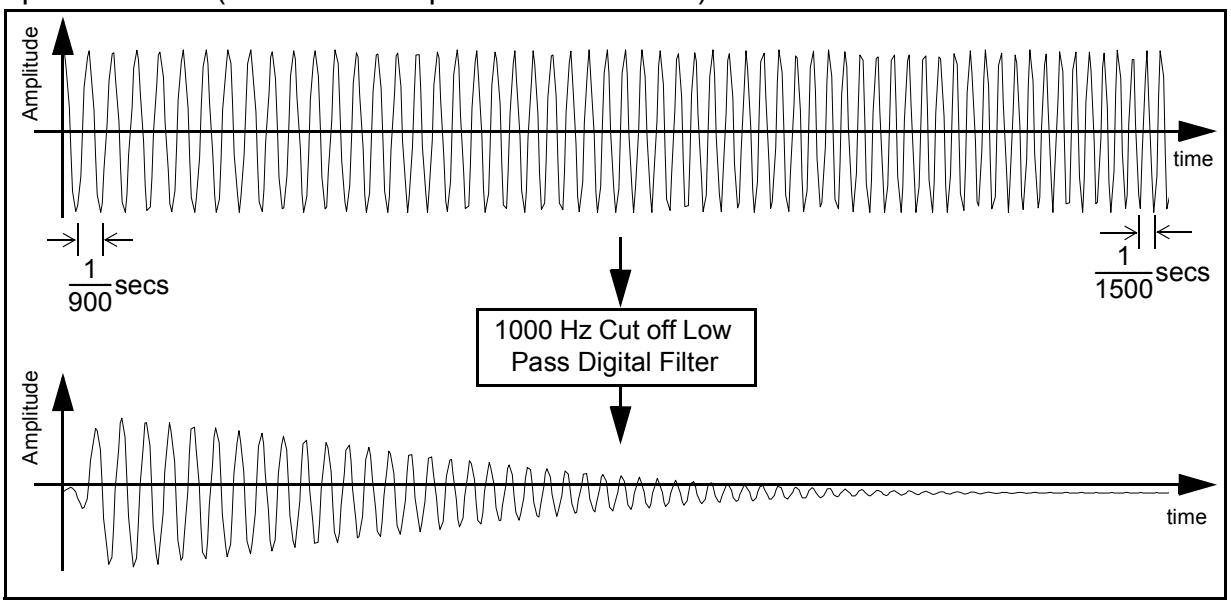
Linear Magnitude Response



Logarithmic Magnitude Response

The 1024 point FFT (zero padded) of the above **DESIGN 1** low pass filter impulse response. The passband ripple is easier to see in the linear plot, whereas the stopband ripple is easier to see in the logarithmic plot.

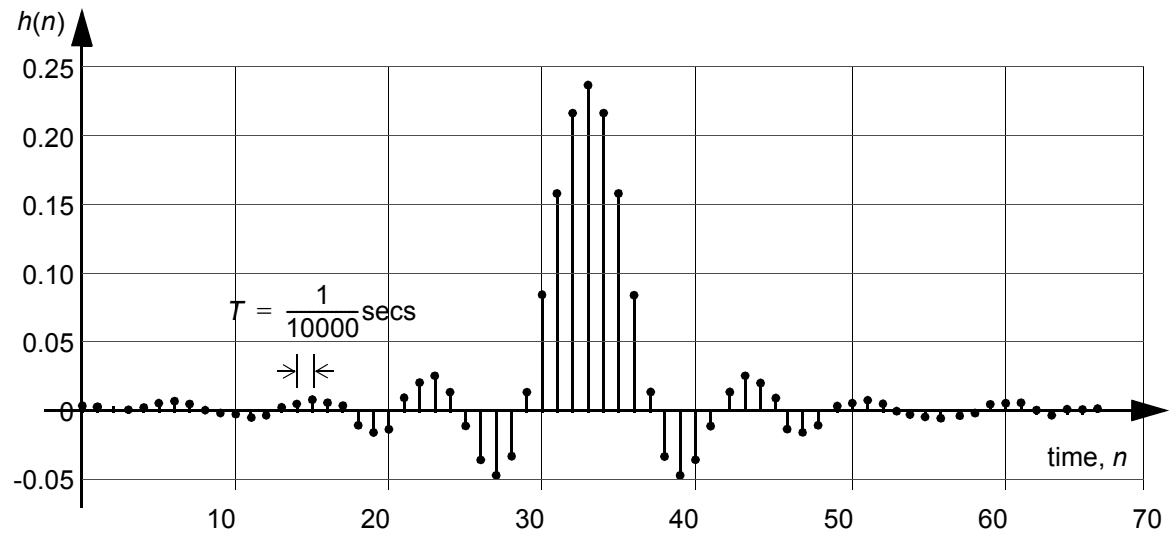
To illustrate the operation of the above digital filter, a chirp signal starting at a frequency of 900 Hz, and linearly increasing to 1500 Hz over 0.05 seconds (500 samples) can be input to the filter and the output observed (individual samples are not shown):



As the chirp frequency reaches about 1000 Hz, the digital filter attenuates the amplitude output signal by a factor of around 0.7 (3dB) until at 1500 Hz the signal amplitude is attenuated by more than 60 dB or a factor of 0.001.

If a low pass filter with less passband ripple and a sharper cut off is required then another filter can be designed, although more weights will be required and the implementation cost of the filter has therefore increased. To illustrate this point, if the above low pass filter is redesigned, but this time with a stopband attenuation of 80dB, a passband ripple of 0.1dB and a transition band of, again,

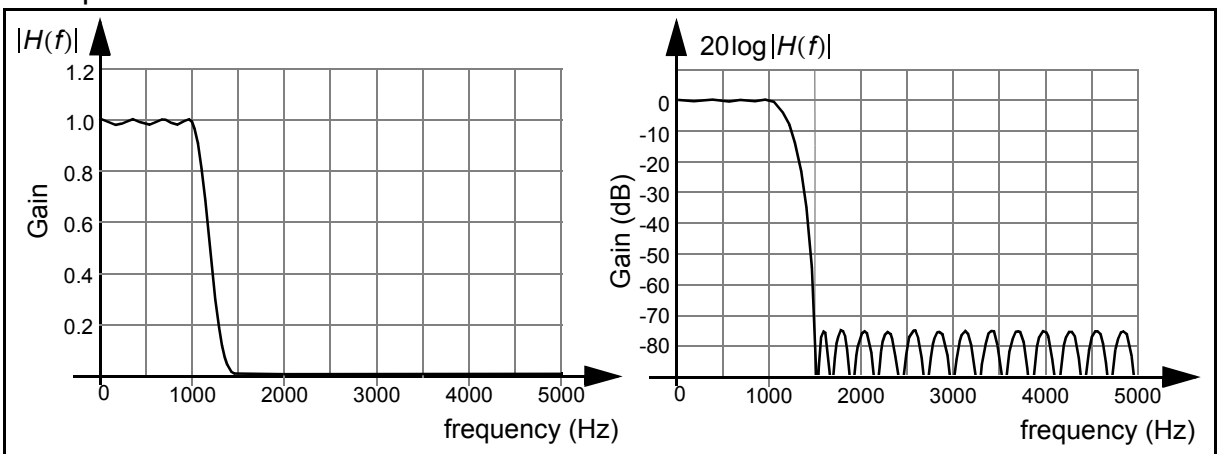
500 Hz, the impulse response of the filter produced by the DSP design software now requires 67 weights:



DESIGN 2: Low Pass FIR Filter Impulse Response

The impulse response $h(n) = w_n$ of a low pass filter with: cut-off frequency 1000 Hz; passband gain 0dB; stopband attenuation 80dB; transition band 500 Hz; passband ripple 0.1dB and sampling at f_s = 10000 Hz. The filter is linear phase and has 67 weights (compare to the above Design 1 which had 31 weights) and therefore an impulse response of duration 67/10000 seconds.

The frequency response of this Design 2 filter can be found by taking the FFT of the *digital filter* impulse response:



Linear Magnitude Response **Logarithmic Magnitude Response**

The 1024 point FFT (zero padded) of the **DESIGN 2** impulse response low pass filter impulse response. Note that, as specified, the filter roll-off is now steeper, the stopband is almost 80 dB and the inband ripple is only fractions of a dB.

Therefore low pass, high pass, bandpass, and bandstop *digital filters* can all be released by using the formal digital filter design methods that are available in a number of DSP software packages. (Or if you have a great deal of time on your hands you can design them yourself with a paper and pencil and reference to one of the classic DSP textbooks!) There are of course many filter design

trade-offs. For example, as already illustrated above, to design a filter with a fast transition between stopband and passband requires more filter weights than a low pass filter with a slow roll-off in the transition band. However the more filter weights, the higher the computational load on the DSP processor, and the larger the group delay through the filter is likely to be. Care must therefore be taken to ensure that the computational load of the *digital filter* does not exceed the maximum processing rate of the DSP processor (which can be loosely measured in multiply-accumulates, MACs) being used to implement it. The minimum computation load of DSP processor implementing a digital filter in the time domain is at least:

$$\text{Computational Load of Digital Filter} = (\text{Sampling Rate} \times \text{No. of Filter Weights}) \text{ MACs} \tag{81}$$

and likely to be a factor greater than 1 higher due to the additional overhead of other assembly language instructions to read data in/out, to implement loops etc. Therefore a 100 weight digital filter sampling at 8000 Hz requires a computational load of 800,000 MACs/second (readily achievable in the mid-1990's), whereas for a two channel digital audio tape (DAT) system sampling at 48kHz and using stereo *digital filters* with 1000 weights requires a DSP processor capable of performing almost 100 million MACs per second (verging on the "just about" achievable with late-1990s DSP processor technology). See also *Adaptive Filter*, *Comb Filter*, *Finite Impulse Response (FIR) Filter, Infinite Impulse Response (IIR) Filter, Group Delay, Linear Phase.*

Digital Filter Order: The order of a digital filter is specified from the degree of the z-domain polynomial. For example, an N weight FIR filter:

$$y(k) = w_0 x(k) + w_1 x(k-1) + \ldots w_{N-1} x(k-N+1) \tag{82}$$

can be written as an N-1th order z-polynomial:

$$\begin{aligned} Y(z) &= X(z)[w_0 + w_1 z^{-1} + w_{N-1} z^{-N+1}] \\ &= X(z) z^{-N+1} [w_0 z^{N-1} + w_1 z^{N-2} + \ldots w_{N-1}] \end{aligned} \tag{83}$$

For an IIR filter, the order of the feedforward and feedback sections of the filter can both be specified. For example an IIR filter with a 0-th order feedforward section (i.e. $N = 1$ above meaning $w_0 = 1$ and all other weights are 0), and an M-1th order feedback section is given by the difference equation:

$$y(k) = x(k) + b_1 y(k-1) + b_2 y(k-2) + \ldots b_{M-1} y(k-M+1) \tag{84}$$

and the M-1th order denominator polynomial is shown below as:

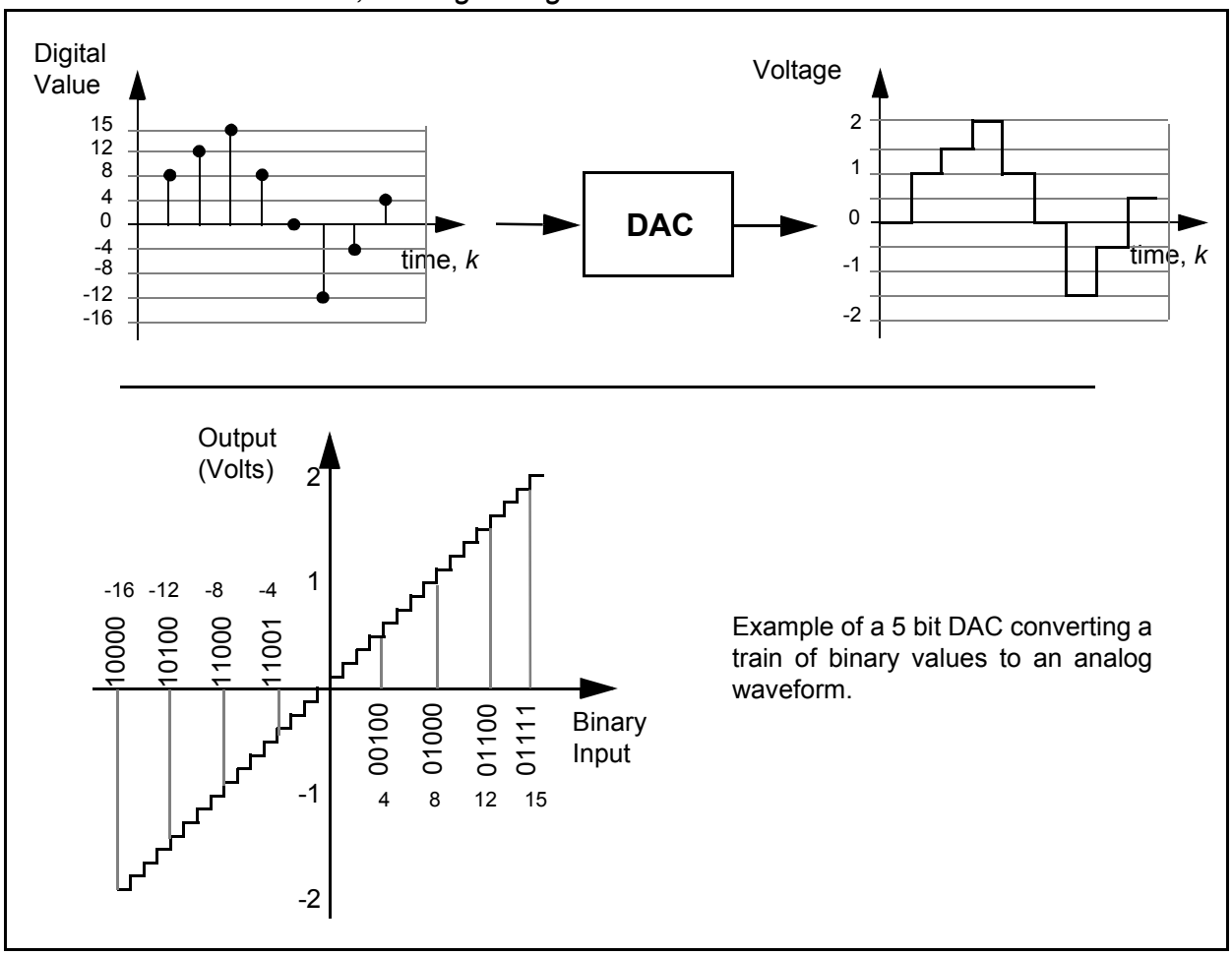
$$\begin{aligned} \frac{Y(z)}{X(z)} &= \frac{1}{1 + b_1 z^{-1} + \ldots + b_{M-2} z^{-M+2} + b_{M-1} z^{-M+1}} \\ &= \frac{z^{M-1}}{z^{M-1} + b_1 z^{M-2} + \ldots + b_{M-2} z + b_{M-1}} \end{aligned} \tag{85}$$

It is worth noting that for an IIR filter the coefficients are indexed starting at 1, i.e. b_1 If a b_0 coefficient were added in the above signal flow graph, then this would introduce a scaling of the output, $y(k)$. See also *Digital Filter, Finite Impulse Response Filter, Infinite Impulse Response Filter*.

Digital Soundfield Processing (DSfP): The name given to the artificial addition of echo and reverberation to a digital audio signal. For example music played in a car can add echo and reverberation to the digital signal prior to being played through the speakers thus giving the impression of the acoustics of a large theatre or a stadium.

Digital Television: The enabling technologies of digital television are presented in detail in [95], [96].

Digital to Analog Converter (D/A or DAC): A digital to analog converter is a device which will take a stream of digital numbers and convert to a continuous voltage signal. Every *digital to analog converter* has an input-output characteristic that specifies the output voltage for a given binary number input. The output of a *DAC* is very *steppy,* and will in fact produce frequency components above the sampling frequency. Therefore a reconstruction filter should be used at the output of a *DAC* to smooth out the steps. Most D/As used in DSP operate using 2's complement arithmetic. See also *Reconstruction Filter, Analog to Digital Converter.*



Example of a 5 bit DAC converting a train of binary values to an analog waveform.

Digital Video Interactive (DVI): Intel Inc. have produced a proprietary digital video compression technology which is generally known as *DVI*. Files that are encoded as *DVI* usually have the suffix, ".dvi" (as do LaTeX^{TM} device independent files -- these are different). See also *Standards*.

Diotic: A situation where the aural stimulation reaching both ears is the same. For example, *diotic* audiometric testing would play the exactly the same sounds into both ears. See also *Audiometry, Dichotic, Monauralic.*

Dirac Impulse or Dirac Delta Function: The continuous time analog to the unit impulse function. See *Unit Impulse Function*.

Direct Broadcast Satellite (DBS): Satellite transmission of television and radio signals may be received directly by a consumer using a (relatively small) parabolic antenna (dish) and a digital tuner. This form of broadcasting is gaining popularity in Europe, Japan, the USA and Australia.

Direct Memory Access: Allowing access to read or write RAM without interrupting normal operation of the processor. The TMS320C40 DSP Processor has 6 independent DMA channels that are 8 bits wide and allow access to memory without interrupting the DSP computation operation. See also *DSP Processor*.

Directivity: A measure of the spatial selectivity of an array of sensors, or a single microphone or antenna. Loosely, *directivity* is the ratio of the gain in the look direction to the average gain in all directions. The higher the *directivity*, the more concentrated the spatial selectivity of a device is in the look direction compared to all other directions. Mathematically, *directivity* is defined for a (power) gain function $G(\theta,\phi,f)$ as:

$$D(f) = \frac{G(0,0,f)}{\frac{1}{4\pi}\int_{FOV} G(\theta,\phi,f)\,d\Omega} \tag{86}$$

where the look direction (and the maximum of the gain function) is assumed to be $\theta=0$ and $\phi=0$ and the field of view *(FOV)* is assumed to be $\Omega = 4\pi$ steradians (units of solid angle). Note that the *directivity* defined above is a function of frequency, *f*, only. If *directivity* as a function of frequency, *D(f)*, is averaged (i.e., integrated) over frequency then a single *directivity* number can be obtained for a wideband system. See also *Superdirectivity, Sidelobe, Main Lobe, Endfire.*

Discrete Cosine Transform (DCT): The DCT is given by the equation:

$$X(k) = \sum_{n=0}^{N-1} x(n)\cos\frac{2\pi kn}{N} \quad \text{for } k = 0 \text{ to } N-1 \tag{87}$$

The DCT is essentially discrete Fourier transform (DFT) evaluated only for the real part of the complex exponential:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{\frac{-j2\pi kn}{N}} \quad \text{for } k = 0 \text{ to } N-1 \tag{88}$$

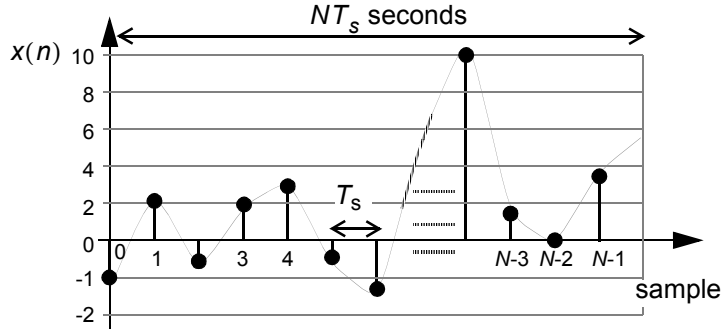
The DCT is used in a number of speech and image coding algorithms. See also Discrete Fourier Transform.

Discrete Fourier Transform: The Fourier transform [57], [58], [93] for *continuous* signals can be defined as:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft}\, df \quad \text{Synthesis}$$

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad \text{Analysis}$$

Fourier Transform Pair (89)



Sampling an analogue signal, $x(t)$, to produce a discrete time signal, $x(nT_s)$ written as $x(n)$. The sampling period is T_s and the sampling frequency is therefore $f_s = 1/T_s$. The total time duration of the N samples is NT_s seconds. Just as there exists a continuous time Fourier transform, we can also derive a discrete Fourier transform (DFT) in order to assess what sinusoidal frequency components comprise this signal.

In the case where a signal is sampled at intervals of T_s seconds and is therefore discrete, the Fourier transform analysis equation will become:

$$X(f) = \int_{-\infty}^{\infty} x(nT_s) e^{-j2\pi fnT_s} d(nT_s) \tag{90}$$

and hence we can write:

$$X(f) = \sum_{n=-\infty}^{\infty} x(nT_0) e^{-j2\pi fnT_0} = \sum_{n=-\infty}^{\infty} x(nT_0) e^{\frac{-j2\pi fn}{f_s}} \tag{91}$$

To further simplify we can write the discrete time signal simply in terms of its sample number:

$$X(f) = \sum_{n=-\infty}^{\infty} x(nT_0) e^{-j2\pi fnT_0} = \sum_{n=-\infty}^{\infty} x(n) e^{\frac{-j2\pi fn}{f_s}} \tag{92}$$

Of course if our signal is causal then the first sample is at $n = 0$, and the last sample is at $n = N-1$, giving a total of N samples:

$$X(f) = \sum_{n=0}^{N} x(n) e^{\frac{-j2\pi fn}{f_s}} \tag{93}$$

By using a finite number of data points this also forces the implicit assumption that our signal is now periodic, with a period of N samples, or NT_s seconds (see above figure). Therefore noting that Eq. 93 is actually calculated for a continuous frequency variable, f, then in actual fact we need only evaluate this equation at specific frequencies which are the zero frequency (DC) and hamonics of the "fundamental" frequency, $f_0 = 1/NT_s = f_s/N$, i.e. $N-1$ discrete frequencies of 0, f_0, $2f_0$, upto f_s.

$$X\left(\frac{kf_s}{N}\right) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kf_s n}{Nf_s}} \quad \text{for } k = 0 \text{ to } N-1 \tag{94}$$

Simplifying to use only the time indice, n, and the frequency indice, k, gives the *discrete Fourier transform:*

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}} \quad \text{for } k = 0 \text{ to } N-1 \tag{95}$$

If we recall that the discrete signal $x(k)$ was sampled at f_s then the signal has image (or alias) components above $f_s/2$, then when evaluating Eq. 95 it is only necessary to evaluate up to $f_s/2$, and therefore the DFT is further simplified to:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}} \quad \text{for } k = 0 \text{ to } N/2 \tag{96}$$

Discrete Fourier Transform

Clearly because we have evaluated the DFT at only N frequencies, then the frequency resolution is limited to the DFT "bins" of frequency width f_s/N Hz.

Note that the discrete Fourier transform only requires multiplications and since each complex exponential is computed in its complex number form.

$$e^{\frac{-j2\pi kn}{N}} = \cos\frac{2\pi kn}{N} - j\sin\frac{2\pi kn}{N} \tag{97}$$

If the signal $x(k)$ is real valued, then the DFT computation requires approximately N^2 real multiplications and adds (noting that a real value multiplied by a complex value requires two real multiplies). If the signal $x(k)$ is complex then a total of $2N^2$ MACs are required (noting that the multiplication of two complex values requires four real multiplications).

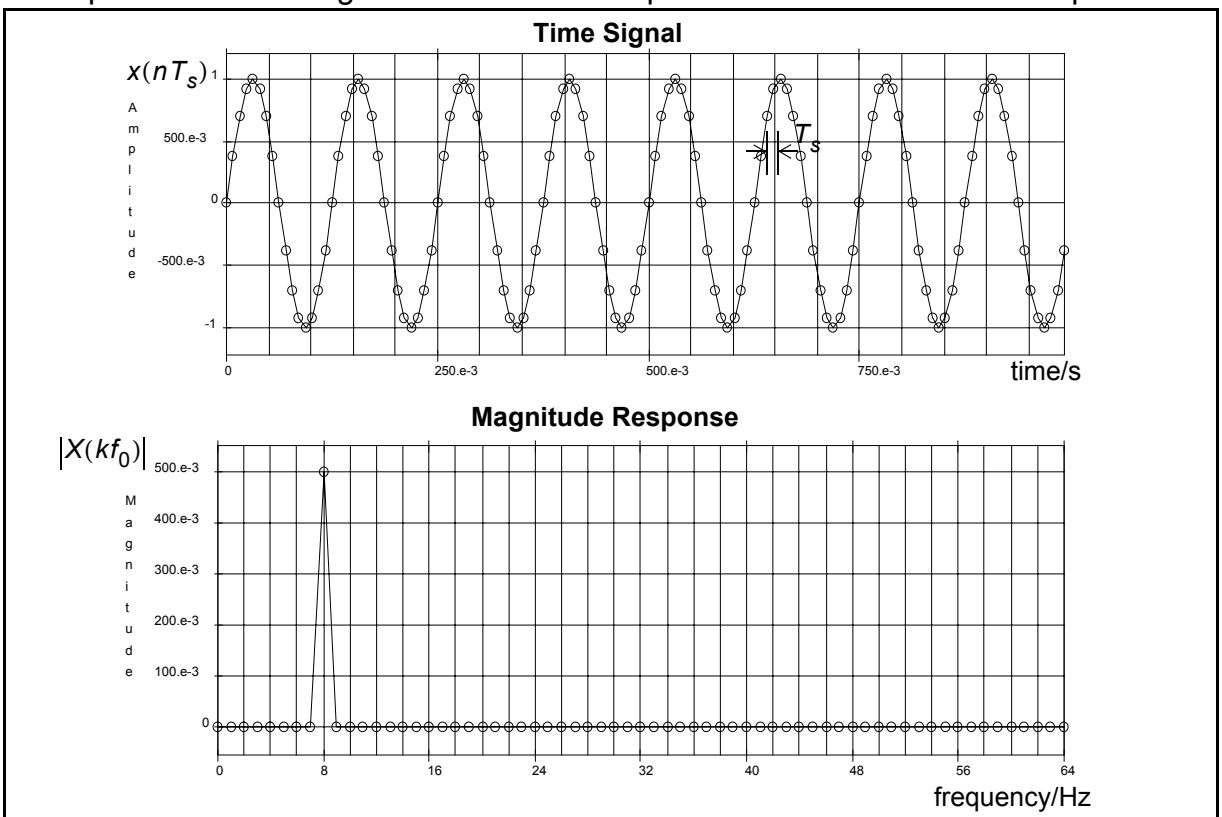
From the DFT we can calculate a magnitude and a phase response:

$$X(k) = |X(k)| \angle X(k) \tag{98}$$

From a given DFT sequence, we can of course calculate the inverse DFT from:

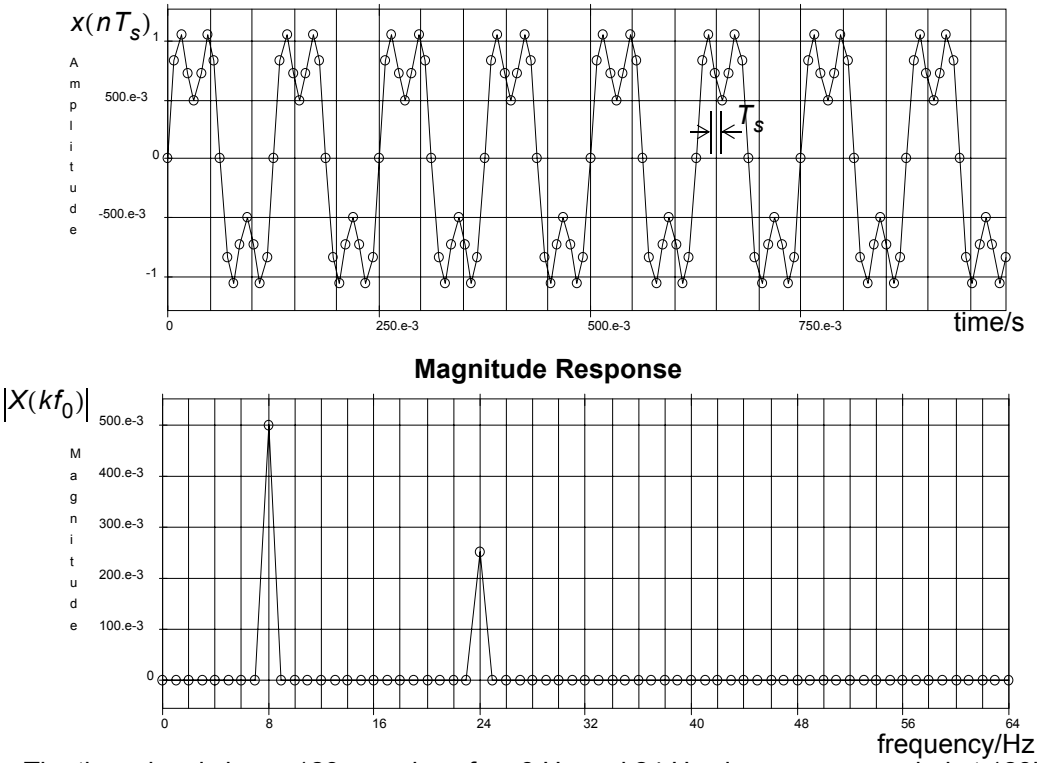
$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{\frac{j2\pi nk}{N}} \tag{99}$$

As an example consider taking the DFT of 128 samples of an 8Hz sine wave sampled at 128 Hz:



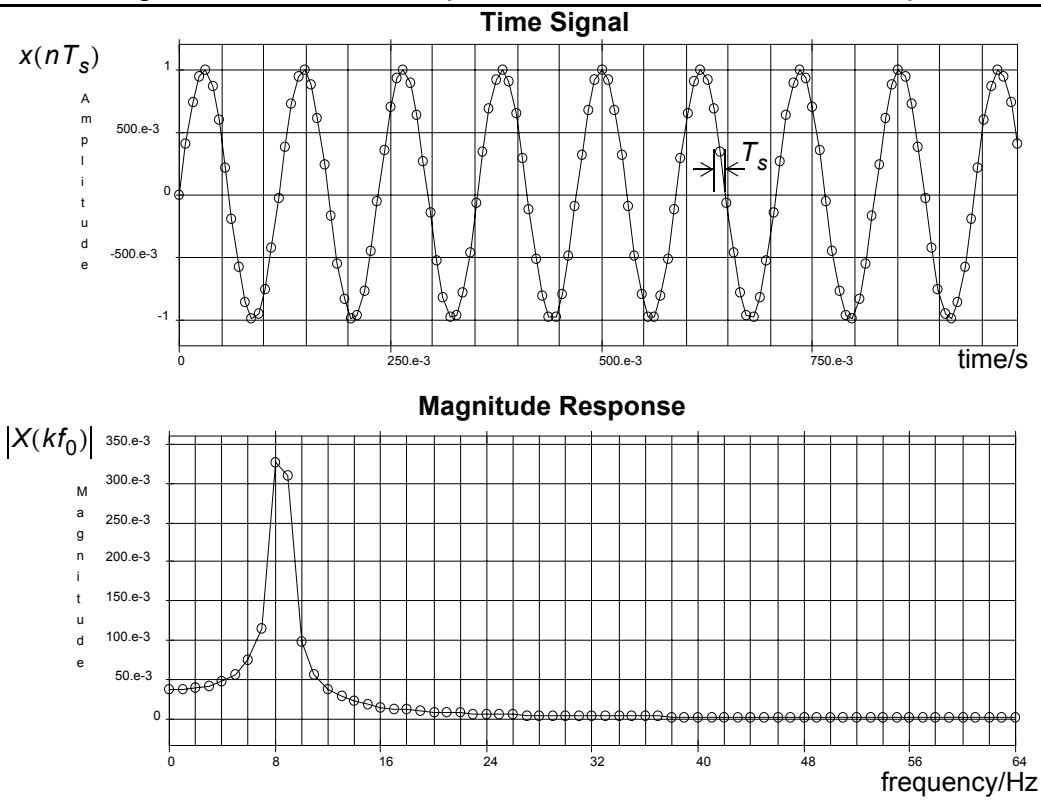
The time signal shows 128 samples of an 8 Hz sine wave sampled at 128Hz: $x(n) = \sin(16\pi n)/128$. Note that there are exactly an integral number of periods (eight) present over the 128 samples. Taking the DFT exactly identifies the signal as an 8 Hz sinusoid. The DFT magnitude spectrum has an equivalent negative frequency portion which is identical to that of the positive frequencies if the time signal was real valued.

If we take the DFT of the slightly more complex signal consisting of an 8Hz and a 24Hz sine wave of half the amplitude of the 8Hz then:



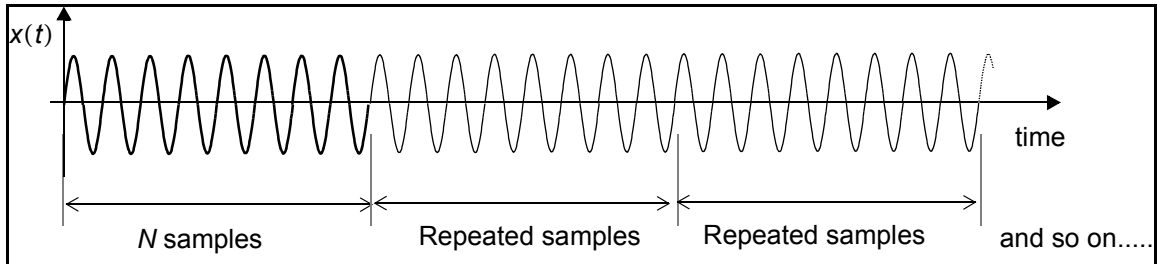
The time signal shows 128 samples of an 8 Hz and 24 Hz sine waves sampled at 128Hz: $x(n) = \sin(16\pi n)/128 + 0.5\sin(48\pi n)/128$. Note that there are exactly an integral number of periods present for both sinusoids over the 128 samples.

Now consider taking the DFT of 128 samples of an 8.5 Hz sine wave sampled at 128 Hz:



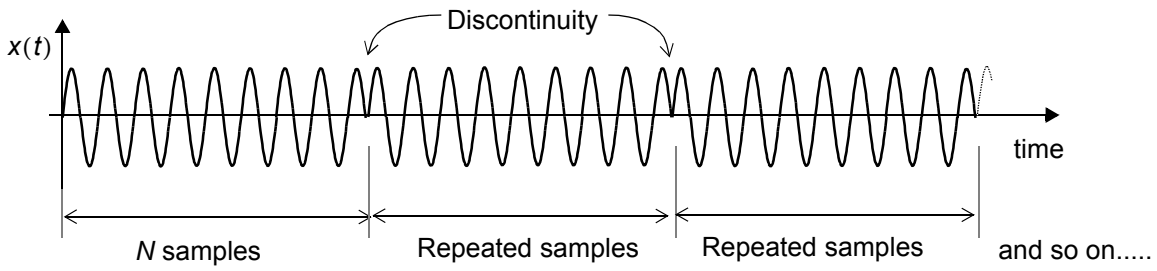
The time signal shows 128 samples of an 8.5 Hz sine wave sampled at 128Hz: $x(n) = \sin(17\pi n)/128$. Note that because the 8.5Hz sine wave does not lie exactly on a frequency bin, then its energy appears spread over a number of frequency bins around 8Hz.

So why is the signal energy now spread over a number of frequency bins? We can interpret this by recalling that the DFT implicitly assumes that the signal is periodic, and the N data points being analysed are one full period of the signal. Hence the DFT assumes the signal has the form:



If there are an integral number of sine wave periods in the N samples input to the DFT computation, then the spectral peaks will fall exactly on one of the frequency bins as shown earlier. Essentially the result produced for the DFT computation has assumed that the signal was periodic, and the N samples form one period of the signal and thereafter the period repeats. Hence the DFT assumes the *complete* signal is as illustrated above (the discrete samples are not shows for clarity.

If there are not an integral number of periods in the signal (as for the 8.5Hz example), then:



If there are not an integral number of sine wave periods in the N samples input to the DFT computation, then the spectral peaks will not fall exactly on one of the frequency bins. As the DFT computation has assumed that the signal was periodic, the DFT interprets that the signal undergoes a "discontinuity" jump at the end of the N samples. Hence the result of the DFT interprets the time signal as if this discontinuity was part of it. Hence more than one single sine wave is required to produce this waveform and thus a number of frequency bins indicate sine wave components being present.

In order to address the problem of spectral leakage, the DFT is often used in conjunction with a windowing function. See also Basis Function, Discrete Cosine Transform, Discrete Fourier Transform - Redundant Computation, Fast Fourier Transform, Fourier, Fourier Analysis, Fourier Series, Fourier Transform, Frequency Response.

Discrete Fourier Transform, Redundant Computation: If we rewrite the form of the DFT in Eq. 96 as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad \text{for } k = 0 \text{ to } N/2 \tag{100}$$

where $W = e^{\frac{j2\pi}{N}}$ Therefore to calculated the DFT of a (trivial) signal with 8 samples requires:

$$\begin{aligned}
X(0) &= x(0) + x(1) + x(2) + x(3) + x(4) + x(5) + x(6) + x(7) \\
X(1) &= x(0) + x(1)W_8^{-1} + x(2)W_8^{-2} + x(3)W_8^{-3} + x(4)W_8^{-4} + x(5)W_8^{-5} + x(6)W_8^{-6} + x(7)W_8^{-7} \\
X(2) &= x(0) + x(1)W_8^{-2} + x(2)W_8^{-4} + x(3)W_8^{-6} + x(4)W_8^{-8} + x(5)W_8^{-10} + x(6)W_8^{12} + x(7)W_8^{-14} \\
X(3) &= x(0) + x(1)W_8^{-3} + x(2)W_8^{-6} + x(3)W_8^{-9} + x(4)W_8^{-12} + x(5)W_8^{-15} + x(6)W_8^{-18} + x(7)W_8^{-21}
\end{aligned} \tag{101}$$

However note that there is redundant computation in Eq. 101. Consider the third term in the second line of Eq. 101:

$$x(2)W_8^{-2} = x(2)e^{j2\pi\left(\frac{-2}{8}\right)} = x(2)e^{\frac{-j\pi}{2}} \tag{102}$$

Now consider the computation of the third term in the fourth line of Eq. 101

$$x(2)W_8^{-6} = x(2)e^{j2\pi\left(\frac{-6}{8}\right)} = x(2)e^{\frac{-j3\pi}{2}} = x(2)e^{j\pi}e^{\frac{-j\pi}{2}} = -x(2)e^{\frac{-j\pi}{2}} \tag{103}$$

There we can save one multiply operation by noting that the term $x(2)W_8^{-6} = -x(2)W_8^{-2}$. In fact because of the periodicity of W_N^{kn} every term in the fourth line of Eq. 101 is available from the terms in the second line of the equation. Hence a considerable saving in multiplicative computations can be achieved. This is the basis of the fast (discrete) Fourier transform discussed under item *Fast Fourier Transform*.

Discrete Fourier Transform, Spectral Aliasing: Note that the discrete Fourier transform of a signal $x(n)$ is periodic in the frequency domain. If we assume that the signal was real and was sampled above the Nyquist rate f_s, then there are no frequency components of interest above $f_s/2$. From the Fourier transform, if we calculate the frequency components up to frequency $f_s/2$ then this is equivalent to evaluating the DFT for the first $N/2-1$ discrete frequency samples:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}} \quad \text{for } k = 0 \text{ to } N/2-1 \tag{104}$$

Of course if we evaluate for the next $N/2-1$ discrete frequencies (i.e. from $f_s/2$ to f_s) then:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}} \quad \text{for } k = N/2 \text{ to } N-1 \tag{105}$$

In Eq. 11 if we substitute for the variable $i = N-k \Rightarrow k = N-i$ and calculate over range $i = 1$ to $N/2$ (equivalent to the range $k = N/2$ to $N-1$) then:

$$X(i) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi in}{N}} \quad \text{for } i = 1 \text{ to } N/2 \tag{106}$$

and we can write:

$$\begin{aligned} X(N-k) &= \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi(N-k)n}{N}} \\ &= \sum_{n=0}^{N-1} x(n) e^{\frac{j2\pi kn}{N}} e^{\frac{-j2\pi Nn}{N}} = \sum_{n=0}^{N-1} x(n) e^{\frac{j2\pi kn}{N}} e^{-j2\pi n} \\ &= \sum_{n=0}^{N-1} x(n) e^{\frac{j2\pi kn}{N}} \quad \text{for } k = N/2 \text{ to } N-1 \end{aligned} \tag{107}$$

since $e^{j2\pi n} = 1$ for all integer values of n. Therefore from Eq. 107 it is clear that:

$$|X(k)| = |X(N-k)| \tag{108}$$

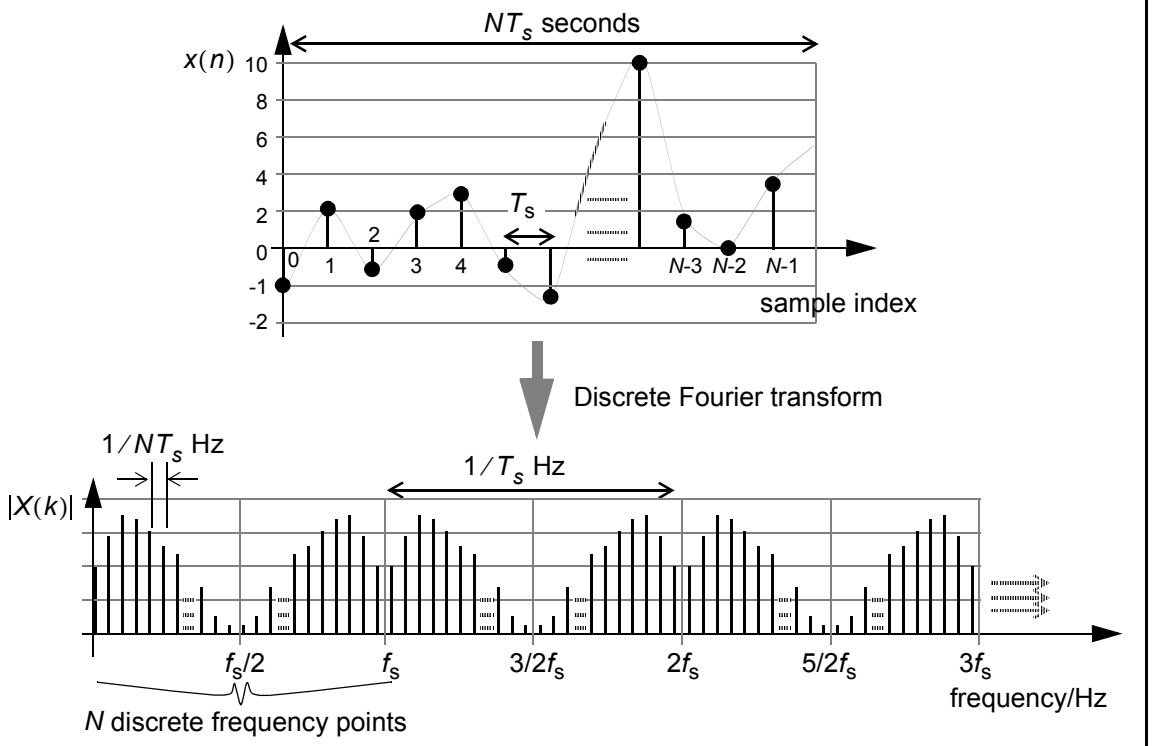
Hence when we plot the DFT it is symmetrical about the $N/2$ frequency sample, i.e. the frequency value $f_s/2$ Hz depending on whether we plot the x-axis as a frequency indice or a true frequency value.

We can further easily show that if we take a value of frequency index k above $N-1$ (i.e. evaluate the DFT above frequency f_s, then:

$$\begin{aligned} X(k+mN) &= \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi(k+mN)n}{N}} = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}} e^{-j2\pi mn} \\ &= \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}} \\ &= X(k) \end{aligned} \tag{109}$$

where m is a positive integer and we note that $e^{j2\pi mn} = 1$.

Therefore we can conclude that when evaluating the magnitude response of the DFT the components of specific interest cover the (baseband) frequencies from 0 to $f_s/2$, and the magnitude spectra will be symmetrical about the $f_s/2$ line and periodic with period f_s:



Spectral aliasing. The main portion of interest of the magnitude response is the "baseband" from 0 to $f_s/2$ Hz. The "baseband" spectra is symmetrical about the point $f_s/2$ and thereafter periodic with period f_s Hz.

See also Discrete Fourier Transform, Fast Fourier Transform, Fast Fourier Transform - Zero Padding, Fourier Analysis, Fourier Series, Fourier Transform.

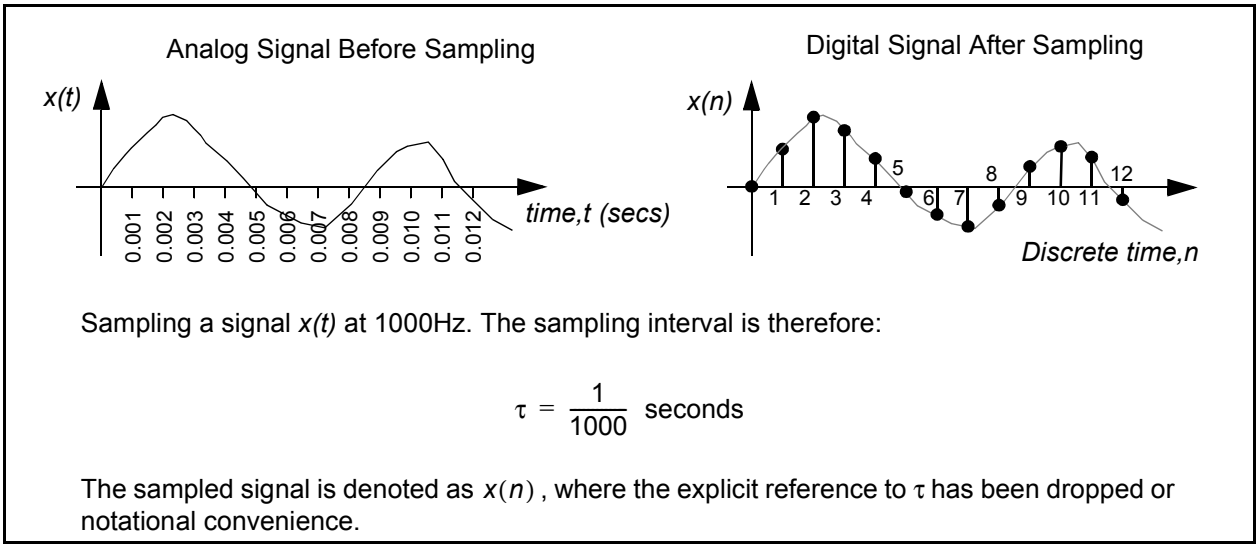
Discrete Time: After an analog signal has been sampled at regular intervals, each sample corresponds to the signal magnitude at a particular *discrete time*. If the sampling period was τ secs, then sampling a continuous time analog signal:

$$x(t) \tag{110}$$

every τ seconds would produce samples

$$x_n = x(n) = x(n\tau), \text{ for } n = 0, 1, 2, 3, \ldots \tag{111}$$

For notational convenience the τ is usually dropped, and only the ***discrete time index***, *n*, is used. Of course, any letter can be used to denote the *discrete time index*, although the most common are: "*n*", "*k*" and "i".



Sampling a signal *x(t)* at 1000Hz. The sampling interval is therefore:

$$\tau = \frac{1}{1000} \text{ seconds}$$

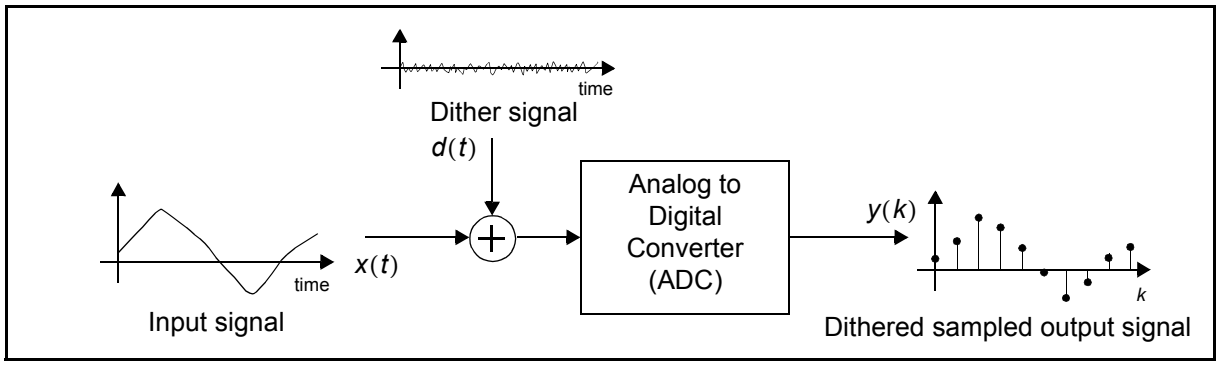
The sampled signal is denoted as $x(n)$, where the explicit reference to τ has been dropped or notational convenience.

Distortion: If the output of a system differs from the input in a non-linear fashion then *distortion* has occurred. For example, if a signal is *clipped* by a DSP system then the output is said to be *distorted*. By the very nature of non-linear functions, a distorted signal will contain frequency components that were not present in the input signal. *Distortion* is also sometimes used to describe linear frequency shaping. See also *Total Harmonic Distortion*.

Distribution Function: See *Random Variable*.

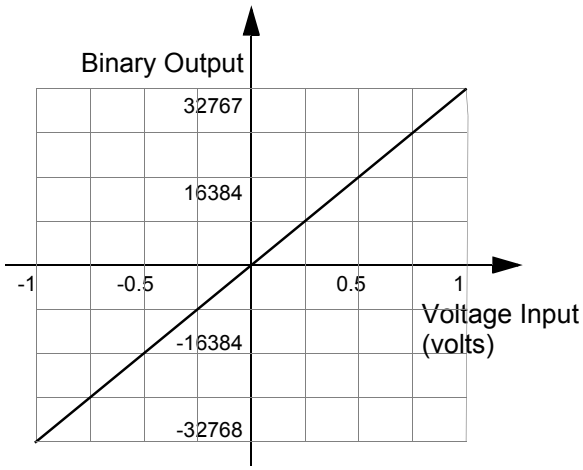
Dithering (audio): *Dithering* is a technique whereby a very low level of noise is added to a signal in order to improve the quality of the psychoacoustically perceived sound. Although the addition of *dithering* noise to a signal clearly reduces the signal to noise ratio (SNR) because it actually adds more noise to the original signal, the overall sound is likely to be improved by breaking up the correlation between the various signal components and quantization error (which, without dithering, results in the quantization noise being manifested as harmonic or tonal distortion).

One form of *dithering* adds a white noise dither signal, $d(t)$ with a power of $q^2/12$, where q is the quantization level of the analog to digital converter (ADC), to the audio signal, $x(t)$ prior to conversion:



Note that without *dithering*, the quantization noise power introduced by the ADC is $q^2/12$, and therefore after *dithering*, the noise power in the digital signal is $q^2/6$, i.e. the noise has doubled or increased by 3dB ($20\log 2$). However the dithered output signal will have decorrelated the quantization error of the ADC and the input signal, thus reducing the harmonic distortion components. This reduction improves the perceived sound quality.

The following example illustrates *dithering*. A 600Hz sine wave of amplitude 6.104×10^{-5} ($= 2/32767$) volts was sampled at 48000Hz with a 16 bit ADC which had the following input/output characteristic:



16 bit Analogue to Digital Converter Input/Output Characteristic.

After analog to digital conversion (with $d(t) = 0$, i.e. no *dithering*) the digital output has an amplitude of 2. On a full scale logarithmic plot, 2 corresponds to -84 dB ($= 20\log(2/32767)$) where

the full scale amplitude of 32767 ($= 2^{15} - 1$) is 0dB. Time and frequency representations of the output of the ADC are shown below, along with a 16384 point FFT of the ADC output:

The frequency representation of the 600Hz sine wave clearly shows that the quantization noise manifests itself as harmonic distortion. Therefore when this signal is reconverted to analog and replayed, the harmonic distortion may be audible.

The magnitude frequency spectrum of the (undithered) signal clearly highlights the tonal distortion components which result from the conversion of this low level signal. The main distortion components are at 1800Hz, 3000Hz, 4200Hz, and so on, (i.e. at 3, 5, 7,..., times the signal's fundamental frequency of 600 Hz).

However if the signal was first dithered by adding an analog white noise *dithering* signal, $d(t)$ of power $q^2/12$ prior to ADC conversion then the time and frequency representations of the ADC output are:

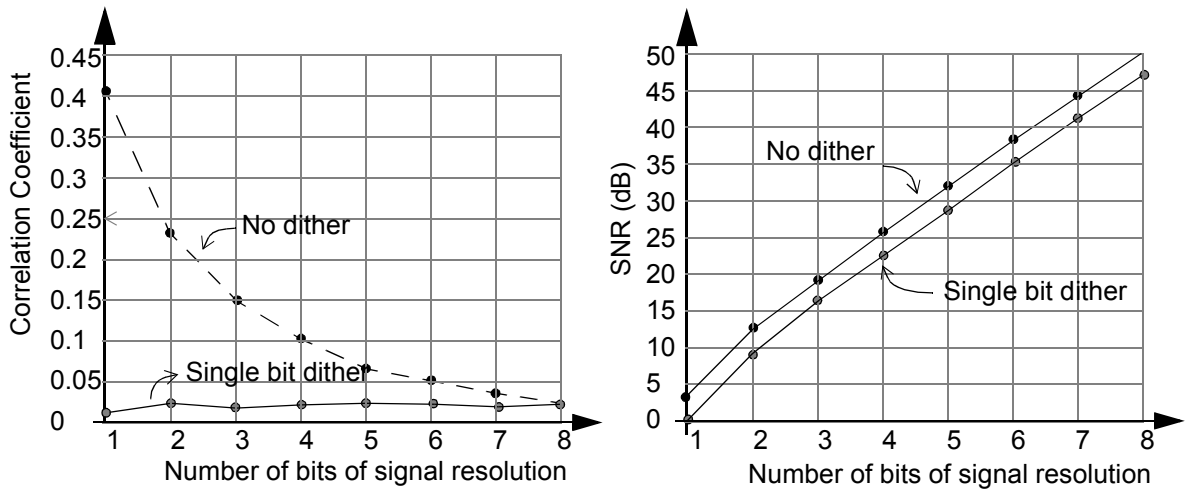
The frequency representation of the dithered 600Hz sine wave clearly shows that the correlation between signal and the quantization error has been removed. Therefore if the signal is reconverted to analog and replayed then the quantization noise is now effectively whitened and harmonic distortion of the signal is no longer perceived.

Note that the magnitude frequency spectrum of the dithered signal has a higher average noise floor, but the tonal nature of the quantization noise has been removed. This dithered signal is more

perceptually tolerable to listen to as the background white noise is less perceptually annoying that than the harmonic noise generated without *dithering*.

Note that a common misconception is that *dithering* can be used to improve the quality of pre-recorded 16 bit hifidelity audio signals. There are, however, no techniques by which a 16 bit CD output can be dithered to remove or reduce harmonic distortion other than add levels of noise to mask it! It may appear in the previous figure as if simply perturbing the quantized values would be a relatively simple and effective dithering technique. There are a number of important differences between dithering before and after the quantizer. First, after the quantizer the noise is simply additive and the spectra of the dither and the harmonically distorted signal add (this is the masking of the harmonic distortion referred to above -- requiring a relatively high power dither). The additive dithering before quantization does not result in additive spectra because the quantization is *nonlinear*. Another difference can be thought of this way: the dither signal is much more likely to cause a change in the quantized level when the input analog signal is close to a quantization boundary (i.e., it does not have to move the signal value very far). After quantization, we have no way of knowing (in the general case) how close an input signal was to a quantization boundary -- so mimicking the dither effect is not, in general, possible. However if a master 20 bit (or higher) resolution recording exists and it is to be remastered to 16 bits, then digital *dithering* is appropriate, whereby the 20 bit signal can be dithered prior to requantizing to 16 bits. The benefits will be similar to those described above for ADCs.

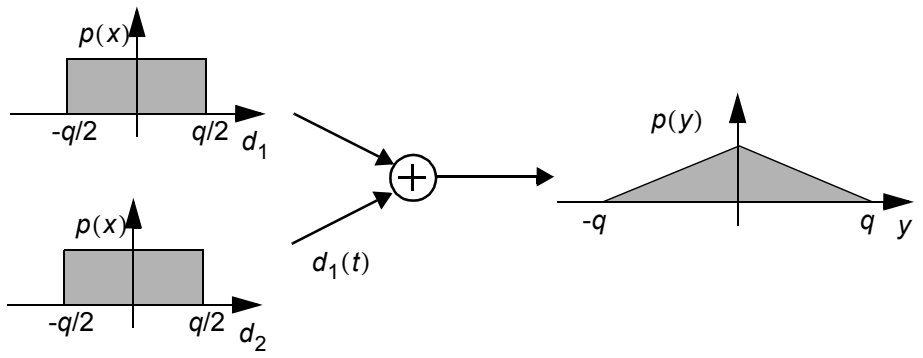
Some simple mathematical analysis of the benefits of *dithering* for breaking up correlation between the signal and the quantization noise can be done. The following figure shows the correlation between a sine wave input signal and the quantization error for 1 to 8 bits of signal resolution:



For low resolution signals the correlation between the signal and quantization error is high. This will be see as tonal or harmonic distortion. however if simple *dithering* scheme is performed prior to analog to digital conversion the correlation can be greatly reduced.

For less than 8 bits resolution the correlation between the signal and quantization noise increases to 0.4 and the signal will sound very (harmonically) distorted. The solid line shows the correlation and signal to noise ratio (SNR) of the signal before and after dither has been added. Clearly the dither is successful at breaking up the correlation between signal and quantization noise and the benefits are greatest for low resolutions. However the total quantization noise in the digital signal after *dithering* is increased by 3dB for all bit resolutions.

A uniformly distributed probability density function (PDF) and maximum amplitude of a half bit ($\pm q/2$) is often used for *dithering*. Adding a single half bit dither signal successfully decorrelates the expected error, however the second moment of the error remains correlated. To decorrelate the second order moment a second uniformly distributed signal can be added. Higher order moments can be decorrelated by adding additional single bits (with uniform probability density functions), however it is found in practice that two uniform random variables (combining to give a triangular probability density function) are sufficient. The effect of adding two random variables with uniform PDFs of $p(x)$ is equivalent to adding a random binary sequence with a triangular PDF (TPDF):



When two uniformly distributed random variables d_1 and d_2, are added together, the probability density function (PDF) of the result, y is a random variable with a triangular PDF (TPDF) obtained by a convolution of the PDFs of d_1 and d_2.

The noise power added to the output signal by one uniform PDF is $q^2/12$, and therefore with two of these *dithering* signals $q^2/6$ noise power is added to the output signal. Noting that the quantization noise power of the ADC is $q^2/12$ and therefore the total noise power of an audio signal dithered with a TPDF is $q^2/4$, i.e. total noise power in the output signal has increased by a factor of 3 or by 4.8 dB ($10\log 3$) over the noise power from the ADC being used without dither. Despite this increase in total noise, the noise power is now more uniformly distributed over frequency (i.e., more white and sounding like a broadband hissing) and the harmonic distortion components caused by correlation between quantization error and the input signal has been effectively attenuated.

In order to mathematically illustrate why dither works, an extreme case of low bit resolution will be addressed. For a single bit ADC (stochastic conversion) the quantizer is effectively reduced to a comparator where:

$$x(k) = \text{sign}(x(k)) = \begin{cases} 1, & v(n) \geq 0 \\ -1, & v(n) < 0 \end{cases} \tag{112}$$

For an input constant (dc) input signal of $v(t) = V_0$ then $x(k) = 1$, if $V_0 > 0$ regardless of the exact magnitude. However by adding a dither signal $d(n)$ with uniform probability density function over the values $Q/2$ and $-Q/2$ before performing the conversion, such that:

$$x(k) = \begin{cases} 1, & v(n) + d(n) \geq 0 \\ -1, & v(n) + d(n) < 0 \end{cases} \tag{113}$$

and taking the mean (expected) value of $x(n)$ gives:

$$E[x(n)] = E[\text{sign}(v(n) + d(n))] = E[\text{sign}(n'(k))] \tag{114}$$

where the $n'(k)$ is a uniformly distributed random variable with a uniform distribution over values of $V_0 - Q/2$ and $V_0 + Q/2$. We can therefore show that the expected or mean value of the dither signal is:

$$\begin{aligned} E[x(n)] &= (-1) \int_{V_0 - Q/2}^{0} \frac{1}{Q} dn' + \int_{0}^{V_0 + Q/2} \frac{1}{Q} dn' \\ &= \frac{1}{Q}\left(V_0 - \frac{Q}{2}\right) + \frac{1}{Q}\left(V_0 + \frac{Q}{2}\right) = \frac{2}{Q} V_0 \end{aligned} \tag{115}$$

Therefore in the mean, the quantizer average dithered output is proportional to V_0. The same intuitive argument can be seen for time varying $x(n)$, as long as the sampling rate is sufficiently fast compared to the changes in the signal.

Dither can be further addressed with oversampling techniques to perform noise shaped *dithering*. See also *Analog to Digital Conversion, Digital to Analog Conversion, Digital Audio, Noise Shaping, Tonal Distortion*.

Divergence: When an algorithm does not converge to a stable solution and instead progresses ever further away from a solution it may be said to be diverging. See also the *Convergence* entry.

Divide and Conquer: The name given to the general problem solving strategy of first dividing the overall problem into a series of smaller sub-problems, solving these subproblems, and finally using the solutions to the subproblems to give the overall solution. Some people also use this as an approach to competing against external groups or managing people within their own organization.

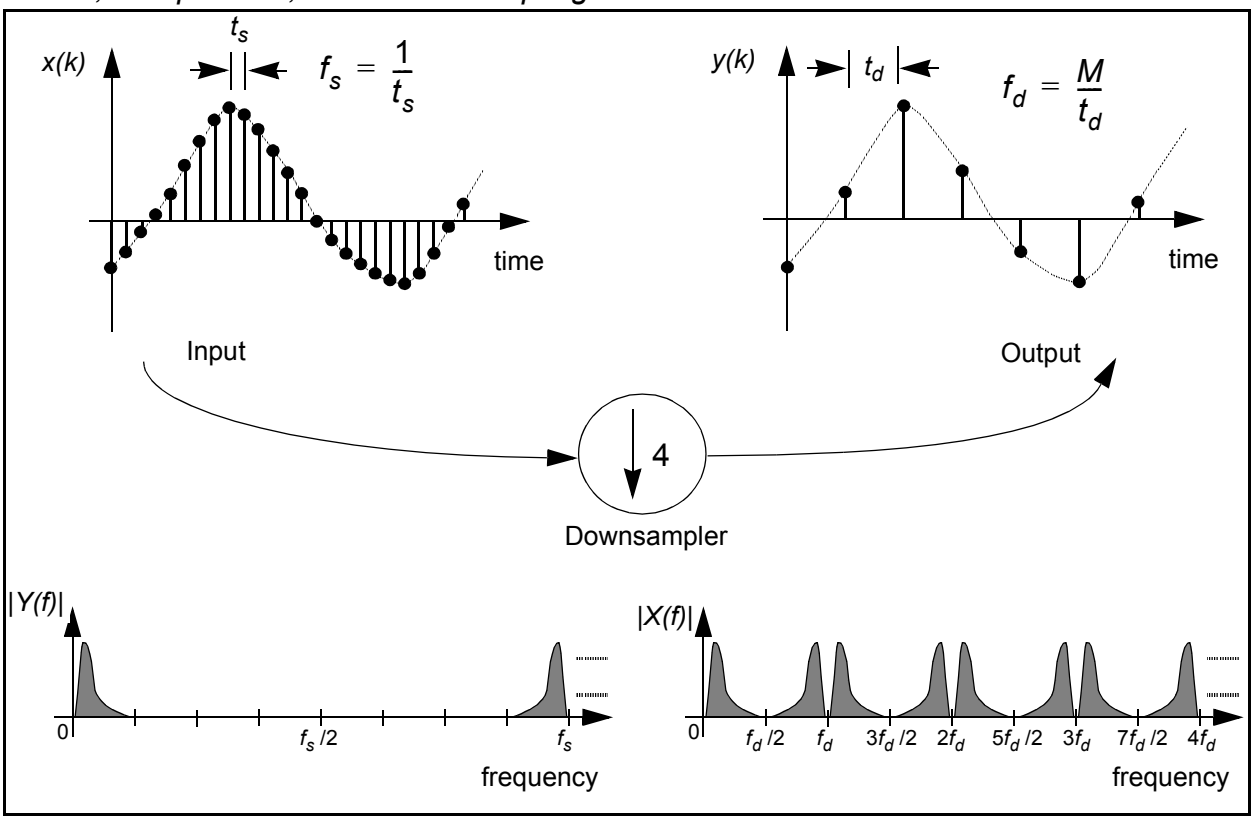
Division: *Division* is rarely required by real time DSP algorithms such as filtering, FFTs, correlation, adaptive algorithms and so on. Therefore DSP processors do not provide a provision for performing fast *division*, in the same way that single cycle parallel multipliers are provided. Therefore *division* is usually performed using a serial algorithm producing a bit at a time result, or using an iterative technique such as Newton-Raphson. Processors such as the DSP56002 can perform a fixed point *division* in around 12 clock cycles. It is worth pointing out however that some DSP algorithms such the QR for adaptive signal processing have excellent convergence and stability properties and do require *division*. Therefore is it possible that in the future some DSP devices may incorporate fast divide and square roots to allow these techniques to be implemented in real time. See also *DSP Processor, Parallel Adder, Parallel Multiplier*.

Dosemeter: See *Noise Dosemeter*.

Dot Product: See *Vector Properties - Inner Product*.

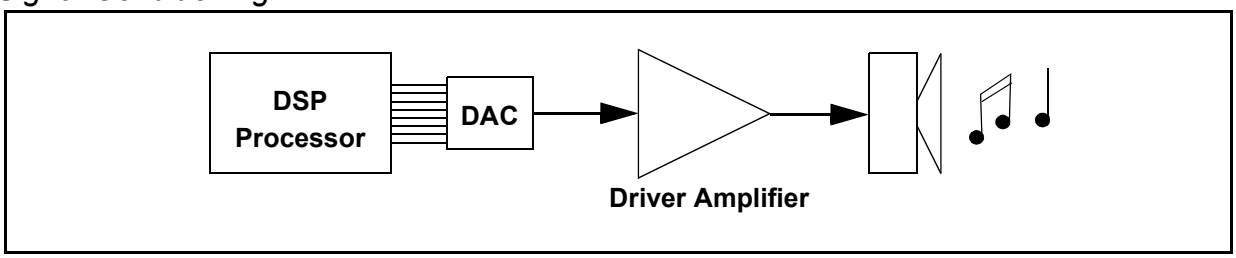
Downsampling: The sampling rate of a digital signal sampled at f_s can be downsampled by a factor of M to a sampling frequency $f_d = f_s/M$ by retaining only every M-th sample. *Downsampling* can lead to aliasing problems and should be performed in conjunction with a low pass filter that cuts-

off at $f_s/2M$; this combination is usually referred to as a decimator. See also *Aliasing, Upsampling, Decimation, Interpolation, Fractional Sampling Rate Conversion*.



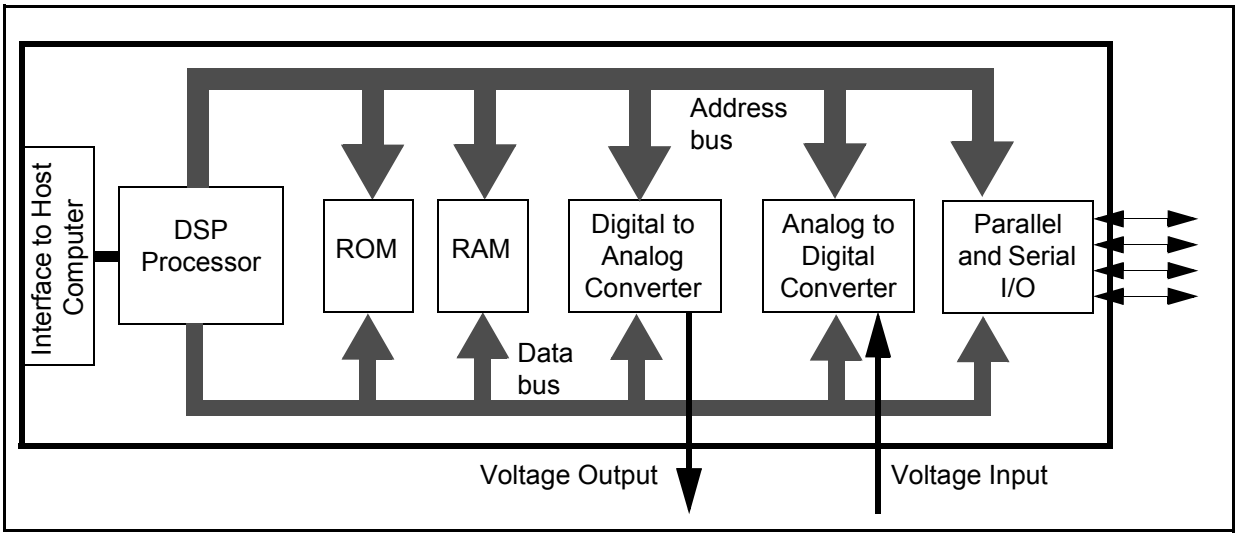
Dr. Bub: The electronic bulletin board operated by Motorola and providing public domain source code, and Motorola DSP related information and announcements.

Driver: The power output from a DAC is usually insufficient to *drive* an actuator such as a loudspeaker. Although the voltage may be at the correct level, the DAC cannot source enough current to deliver the required power. Therefore a *driver* in the form of an amplifier is required. See also *Signal Conditioning*.

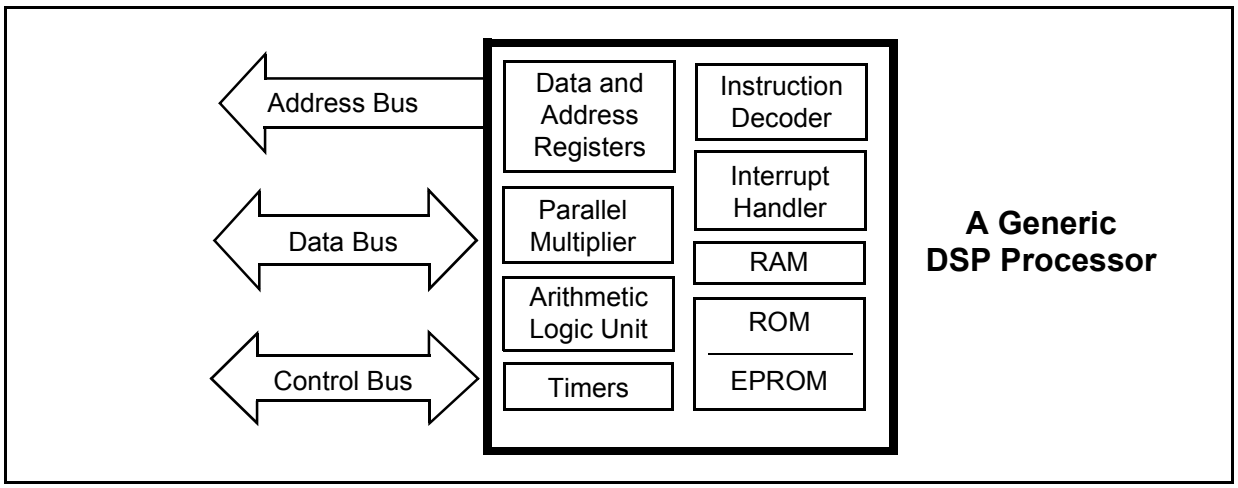


DSP Board: A DSP board is a generic name for a printed circuit board (PCB) which has a DSP processor, memory, A/D and D/A capabilities, and digital input ports (parallel and serial). For development work most DSP boards are plug-in modules for computers such as the IBM-PC, and Macintosh. The computer is used as a host to allow assembly language programs to be conveniently developed and tested using assemblers and cross compilers. When an application

has been fully developed, a stand-alone *DSP board* can be realized. See also *Daughter Module, DSP Processor, Motherboard*.



DSP Processor: A microprocessor that has been designed for implementing DSP algorithms. The main features of these chips are fast interrupt response times, a single cycle parallel multiplier, and a subset of the assembly language instructions found on a general purpose microprocessor (e.g. Motorola 68030) to save on silicon area and optimize DSP type instructions. The main *DSP processors* are the families of the DSP56/96 (Motorola), TMS320 (Texas Instruments), ADSP 2100 (Analog Devices), and DSP16/32 (AT&T). DSP Processors are either floating point or fixed point devices. See also *DSP Board*.

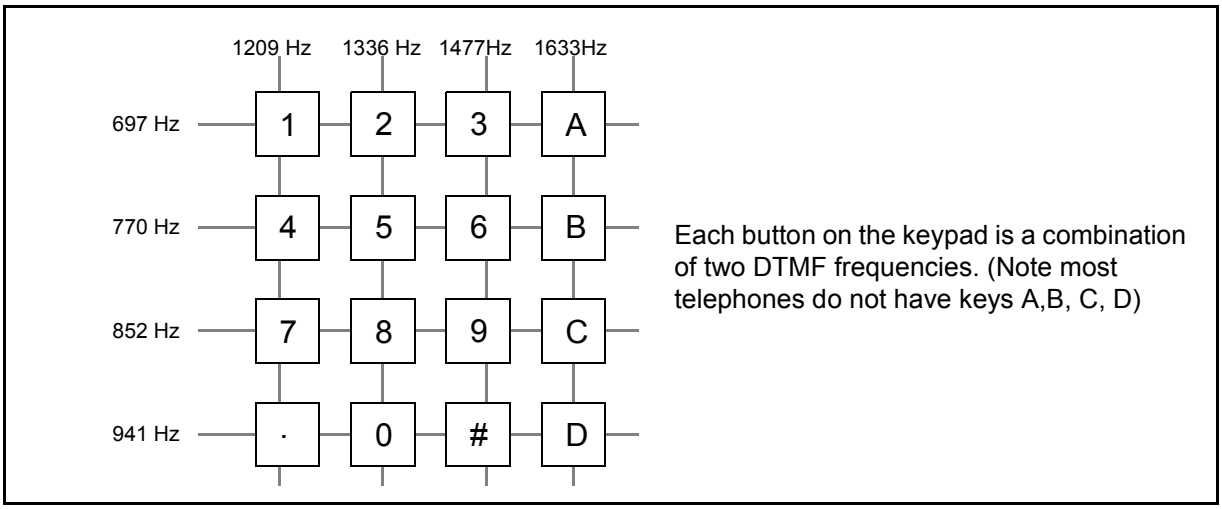


DSPLINK$_{TM}$: A bidirectional and parallel 16 bit data interface path used on Loughborough Sound Images Ltd. (UK) and Spectron (USA) DSP boards to allow high speed communication between separate DSP boards and peripheral boards. The use of DSPLINK means that data between separate boards in a PC do not need to communicate data via the PC bus.

Dual: A prefix to mean “two of”. For example the Burr Brown DAC2814 chip is described as a Dual 12 Bit Digital to Analog Converter (DAC) meaning that the chip has two separate (or independent) DACs. In the case of DACs and ADCs, if the device is used for hi-fidelity audio *dual* devices are often referred to as stereo. See also *Quad*.

Dual Slope: A type of A/D converter.

Dual Tone Multifrequency (DTMF): *DTMF* is the basis of operation of push button tone dialing telephones. Each button on a touch tone telephone is a combination of two frequencies, each from a group of four. $2^4 = 16$ possible combinations of tones pairs can be encoded using the two groups of four tones. The two groups of four frequencies are: (low) 697Hz, 770Hz, 852Hz, 941Hz, and (high) 1209Hz, 1336Hz, 1477Hz, and 1633Hz:



The standards for *DTMF* signal generation and detection are given in the ITU (International Telecommunication Union) standards Q.23 and Q.24. In current telephone systems, virtually every telephone now uses *DTMF* signalling to allow transmission of a 16 character alphabet for applications such as number dialing, data entry, voice mail access, password entry and so on. The *DTMF* specifications commonly adopted are:

Signal Frequencies:

- Low Group 697, 770, 852, 941 Hz
- High Group: 1209, 1336, 1477, 1633 Hz

Frequency tolerance:

- Operation: ≤ 1.5%

Power levels per frequency:

- Operation: 0 to -25dBm
- Non-operation: -55dBm max

Power level difference between frequencies

- +4dB to -8dB

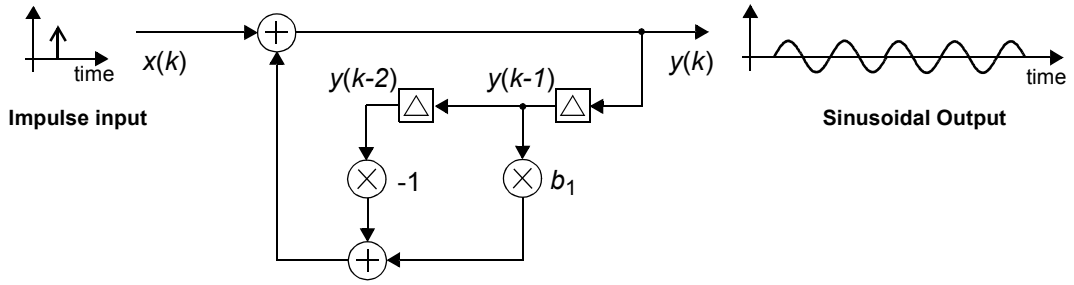
Signal Reception timing:

- Signal duration: operation: 40ms (min)
- Signal duration: non-operation: 23ms (max)
- Pause duration: 40ms (min);
- Signal interruption: 10ms (max);
- Signalling velocity: 93 ms/digit (min).

See also *Dual Tone Multifrequency - Tone Detection*, *Dual Tone Multifrequency - Tone Generation*, *Goertzel's Algorithm*.

Dual Tone Multifrequency (DTMF), Tone Generation: One method to generate a tone is to use a sine wave look up table. For example some members of the Mototola DSP56000 series of processors include a ROM encoded 256 element sine wave table which can be used for this purpose. Noting that each DTMF signal is a sum of two tones, then it should be possible to use a look up table at different sampling rates to produce a DTMF tone.

An easier method is to design a "marginally stable" IIR (infinite impulse response) filter whereby the poles of the filter are on the unit circle and the filter impulse response is a sinusoid at the desired frequency. This method of tone generation requires only a few lines of DSP code, and avoids the requirement for "expensive" look-up tables. The structure of an IIR filter suitable for tone generation is simply:



A two pole "marginally stable" IIR filter. For an input of an impulse the filter begins to oscillate.

This operation of this 2 pole filter can be analysed by considering the z-domain representation. The discrete time equation for this filter is:

$$y(k) = x(k) + \sum_{n=1}^{2} b_n y(k-n) = x(k) + by(k-1) - y(k-2) \tag{116}$$

where we now write $b_1 = b$ and $b_2 = -1$. Writing this in the z-domain gives:

$$Y(z) = X(z) + bz^{-1}Y(z) - z^{-2}Y(z) \tag{117}$$

The transfer function, $H(z)$, is therefore:

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} \\ &= \frac{1}{1 - bz^{-1} + z^{-2}} = \frac{1}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})} = \frac{1}{1 - (p_1 + p_2)z^{-1} + p_1 p_2 z^{-2}} \end{aligned} \tag{118}$$

where, p_1 and p_2 are the poles of the filter, and $b = p_1 + p_2$ and $p_1 p_2 = 1$. The poles of the filter, $p_{1,2}$ (where the notation $p_{1,2}$ means p_1 and p_2) can be calculated from the quadratic formula as:

$$p_{1,2} = \frac{b \pm \sqrt{b^2 - 4}}{2} = \frac{b \pm j\sqrt{4 - b^2}}{2} \tag{119}$$

Given that b is a real value, then p_1 and p_2 are complex conjugates. Rewriting Eq. 119 in polar form gives:

$$p_{1,2} = e^{\pm j \tan^{-1} \frac{\sqrt{4 - b^2}}{b}} \tag{120}$$

Considering the denominator polynomial of Eq. 118, the magnitude of the complex conjugate values p_1 and p_2 are necessarily both 1, and the poles will lie on the unit circle. In terms of the frequency placement of the poles, noting that this is given by:

$$\left| p_{1,2} \right| = 1 = e^{\frac{\pm j 2\pi f}{f_s}} \tag{121}$$

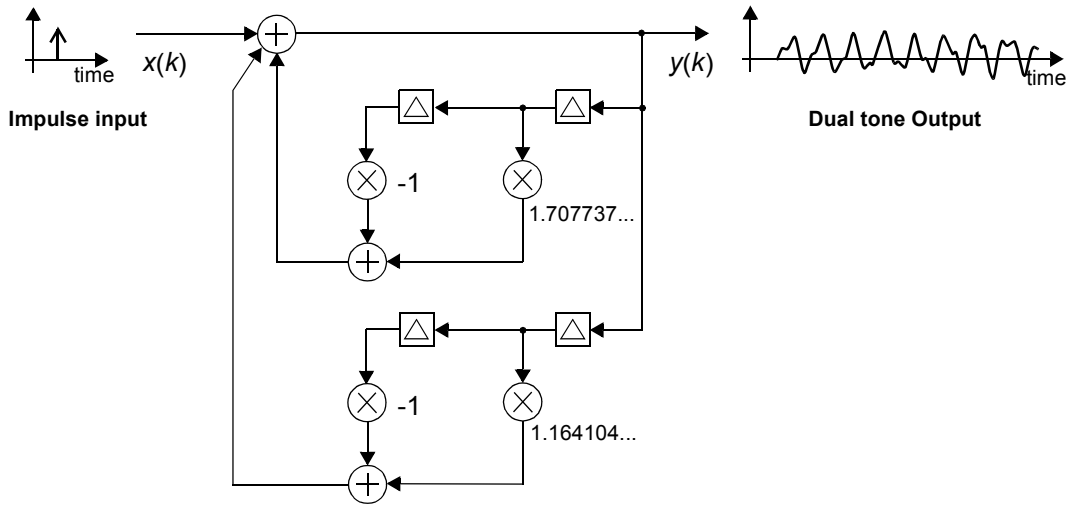
(where $\left| e^{j\omega} \right| = 1$ for any ω) for a sampling frequency f_s, from Eqs. 121 and 120 it follows that:

$$\frac{2\pi f}{f_s} = \tan^{-1} \frac{\sqrt{4 - b^2}}{b} \tag{122}$$

For most telecommunication systems the sampling frequency is $f_s = 8000\,\text{Hz}$. The values of b for the various desired *DTMF* frequency of oscillations can therefore be calculated from Eq. 122 to be:

b	frequency, f / Hz
1.707737809	697
1.645281036	770
1.568686984	852
1.478204568	941
1.164104023	1209
0.996370211	1336
0.798618389	1477
0.568532707	1633

For example, in order to generate the *DTMF* signal for the digit #1, it is required to produce two tones, one at 697 Hz and one at 1209 Hz. This can be accomplished by using the IIR filter :



An IIR filter to produce the *DTMF* signal for the digit #1. The filter consists of two "marginally stable" two pole IIR files producing the 697 Hz tone (top) and the 1209 Hz tone (bottom) added together. Note that the filters will have different magnitude responses and therefore the two tones are unlikely to have the same amplitude. The ITU standard allows for this amplitude difference.

See also *Dual Tone Multifrequency (DTMF) - Tone Detection, Dual Tone Multifrequency (DTMF) - Tone Detection, Goertzel's Algorithm*.

Dual Tone Multifrequency (DTMF), Tone Detection: DTMF tones can be detected by performing a discrete Fourier transform (DFT), and considering the level of power that is present in a particular frequency bin. Because DTMF tones are often used in situations where speech may also be present, it is important that any detection scheme used can distinguish between a tone and a speech signal that happens to have strong tonal components at a DTMF frequency. Therefore for a DTMF tone at f Hz, a detection scheme should check for the signal component at f Hz and also check that there is *no* discernable component at $2f$ Hz; quasi-periodic speech components (such as vowel sounds) are rich in (even) harmonics, whereas DTMF tones are not.

The number of samples used in calculating the DFT should be shorter than the number of samples in half of a DTMF signalling interval, typically of 50ms duration equivalent to 400 samples at a sampling frequency of $f_s = 8000$ Hz, but be large enough to give a good frequency resolution. The DTMF standards of the International Telecommunication Union (ITU) therefore suggest a value of 205 samples in standards Q.23 and Q.24. Using this 205 point DFT the DTMF fundamental and the second harmonics of the 8 possible tones can be successfully discerned. Simple decision logic is applied to the DFT output to specify which tone is present. The second harmonic is also detected in order that the tones can be discriminated from speech utterances that happen to include a frequency component at one of the 8 frequencies. Speech can have very strong harmonic content, whereas the DTMF tone will not. To add robustness against noise, the same DTMF tones require to be detected in a row to give a valid DTMF signal .

If a 205 point DFT is used, then the frequency resolution will be:

$$\text{Frequency Resolution} = \frac{8000}{205} = 39.02 \text{ Hz} \tag{123}$$

The DTMF tones therefore do not all lie exactly on the frequency bins. For example the tone at 770 Hz will be detected at the frequency bin of 780 Hz (20×39.02 Hz). In general the frequency bin, k to look for a single tone can be calculated from:

$$k = \text{int}\left(\frac{f_{\text{tone}} N}{f_s}\right) \tag{124}$$

where f_{tone} is a DTMF frequency, $N = 205$ and $f_s = 8000$ Hz. The bins for all of the DTMF tones for these parameters are therefore:

frequency, *f* / Hz	*bin*
697	18
770	20
852	22
941	24
1209	31
1336	34
1477	38
1633	42

When the 2nd harmonic of a DTMF frequency is to be considered, then the bin at twice the fundamental frequency bin value is detected (there should be no appreciable signal power there for a DTMF frequency). When calculating the DFT for DTMF detection because we are only interested in certain frequencies, then it is only necessary to calculate the frequency components at the frequency bins of interest. Therefore an efficient algorithm based on the DFT called Goertzel's algorithm is usually used for DTMF tone detection. See also *Dual Tone Multifrequency , Dual Tone Multifrequency - Tone Generation*, *Goertzel's Algorithm*.

Dynamic Link Library: A library of compiled software routines in a separate file on disk that can be called by a Microsoft Windows program.

Dynamic RAM (DRAM): Random access memory that needs to be periodically refreshed (electrically recharged) so that information that is stored electrically is not lost. See also *Non-volatile RAM, Static RAM*.

Dynamic Range: *Dynamic range* specifies the numerical range, giving an indication of the largest and smallest values that can be correctly represented by a DSP system. For example if 16 bits are used in a system then the linear (amplitude) *dynamic range* is $-2^{15} \rightarrow 2^{15}-1$ (-32768 to +32767). Usually *dynamic range* is given in decibels (dB) calculated from 20 $\log_{10}$ (Linear Range), e.g. for 16 bits $20\log_{10}2^{16}$ =96dB.

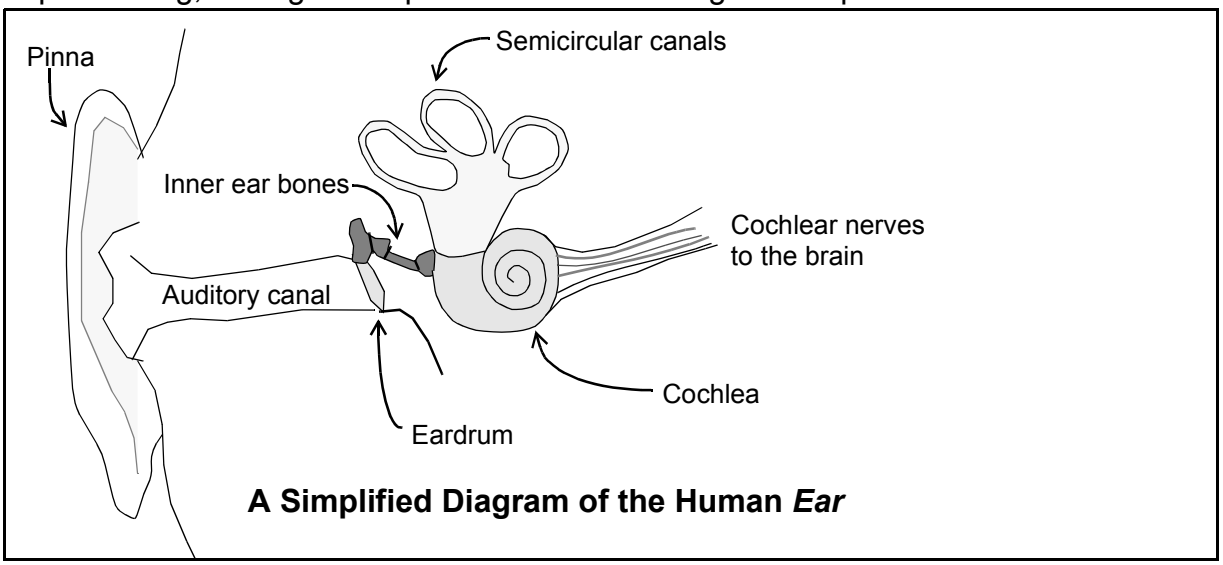
E

***e*:** The natural logarithm base, $e = 2.7182818\ldots$. e can be derived by taking the following limit:

$$e \equiv \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n \tag{125}$$

See also *Exponential Function.*

Ear: The ear is a basically the system of flesh, bone, nerves and brain allowing mammals to perceive and react to sound. It is probably fair to say that a very large percentage of DSP is dealing with the processing, coding and reproduction of audio signals for presentation to the human *ear*.



A Simplified Diagram of the Human *Ear*

The human *ear* can be generally described as consisting of three parts, the outer, middle and inner *ear*. The outer *ear* consists of the pinna and the *ear* canal. The shape of the external *ear* has evolved such that is has good sensitivity to frequencies in the range 2 - 4kHz. Its complex shape provides a number of diffracted and reflected acoustic paths into the middle *ear* which will modify the spectrum of the arriving sound. As a result a single *ear* can actually discriminate direction of arrival of broadband sounds.

The *ear* canal leads to the *ear* drum (tympanic membrane) which can flex in response to sound. Sound is then mechanically conducted to the inner *ear* interconnection of bones (the ossicles), the malleus (hammer), the incus (anvil) and the stapes (stirrup) which act as an impedance matching network (with the ear drum and the oval window of the cochlea) to improve the transmission of acoustic energy to the inner *ear*. Muscular suppression of the ossicle movement provides for additional compression of very loud sounds.

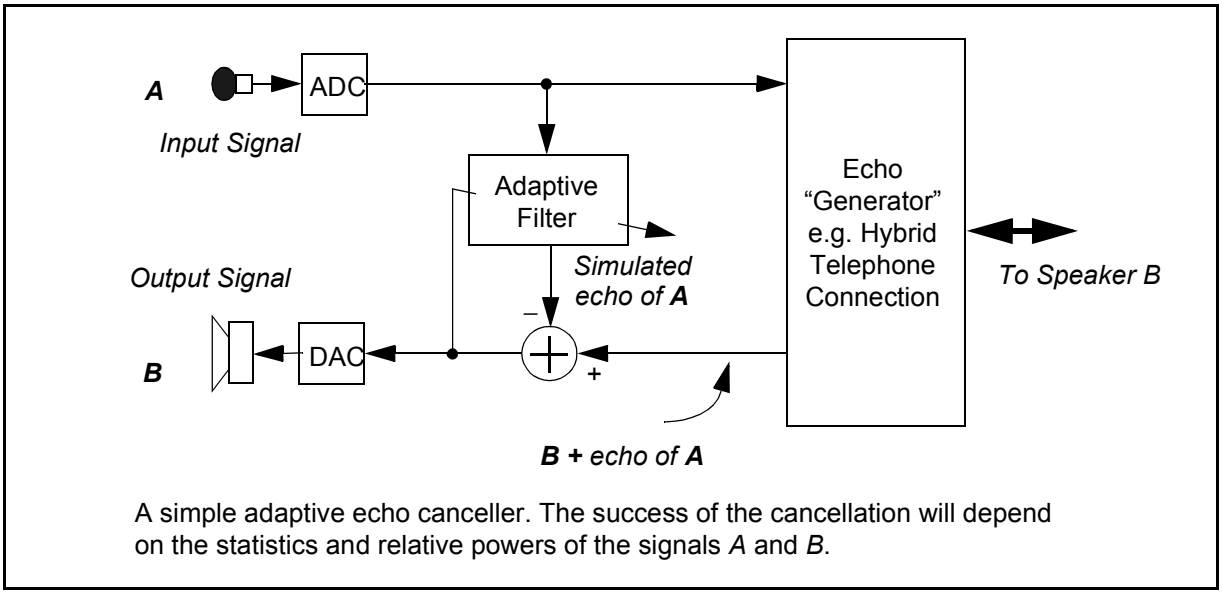
The inner *ear* consists mainly of the cochlea and the vestibular system which includes the semicircular canals (these are primarily used for balance). The cochlea is a fluid filled snail-shell shaped organ that is divided along its length by two membranes. Hair cells attached to the basilar membrane detect the displacement of the membrane along the distance from the oval window to the end of the cochlea. Different frequencies are mapped to different spots along the basilar membrane. The further the distance from the oval window, the lower the frequency. The basilar membrane and its associated components can be viewed as acting like a series of bandpass filters

sending information to the brain to interpret [30]. In addition, the output of these filters is logarithmically compressed. The combination of the middle and inner ear mechanics allows signals to be processed over the amazing dynamic range of 120dB. See also See also *Audiology, Audiometer, Audiometry, Auditory Filters, Hearing Impairment, Threshold of Hearing.*

EBCDIC: See also *ASCII.*

Echo: When a sound is reflected of a nearby wall or object, this reflection is called an *echo*. Subsequent *echoes (of echoes)*, as would be clearly heard in a large, empty room are referred to collectively as reverberations. *Echoes* also occur on telecommunication systems where impedance mismatches reflect a signal back to the transmitter. *Echoes* can sometimes be heard on long distance telephone calls. See also *Echo Cancellation*, *Reverberation*.

Echo Cancellation: An *echo canceller* can be realised [53] with an adaptive signal processing system identification architecture. For example if a telephone line is causing an echo then by incorporating an adaptive *echo canceller* it should be possible to attenuate this echo:



A simple adaptive echo canceller. The success of the cancellation will depend on the statistics and relative powers of the signals *A* and *B*.

When speaker A (or data source A) sends information down the telephone line, mismatches in the telephone hybrids can cause echoes to occur. Therefore speaker A will hear an echo of their own voice which can be particularly annoying if the echo path from the near and far end hybrids is particularly long. (Some echo to the earpiece is often desirable for telephone conversation, and the local hybrid is deliberately mismatched. However for data transmission echo is very undesirable and must be removed.) If the echo generating path can be suitably modelled with an adaptive filter, then a negative simulated echo can be added to cancel out the signal A echo. At the other end of the line, telephone user B can also have an echo canceller.

In general local echo cancellation (where the adaptive *echo canceller* is inside the consumer's telephone/data communication equipment) is only used for data transmission and not speech. Minimum specifications for the ITU V-series of recommendations can be found in the CCITT Blue Book. For V32 modems (9600 bits/sec with Trellis code modulation) an echo reduction ratio of 52dB is required. This is a power reduction of around 160,000 in the echo. Hence the requirement for a powerful DSP processor.

For long distance telephone calls where the round trip echo delay is more than 0.1 seconds and suppressed by less than 40dB (this is typical via satellite or undersea cables) line echo on speech

can be a particularly annoying problem. Before adaptive *echo cancellers* were cost effective to implement, the echo problem was be solved by setting up speech detectors and allowing speech to be half duplex. This was inconvenient for speakers who were required to take turns speaking. Adaptive *echo cancellers* at telephone exchanges have helped to solve this problem. The set up of the telephone exchange *echo cancellers* is a little different from the above example and the echo is cancelled on the outgoing signal line, rather than the incoming signal line. See also *Acoustic Echo Cancellation, Adaptive Filtering, Least Mean Squares Algorithm.*

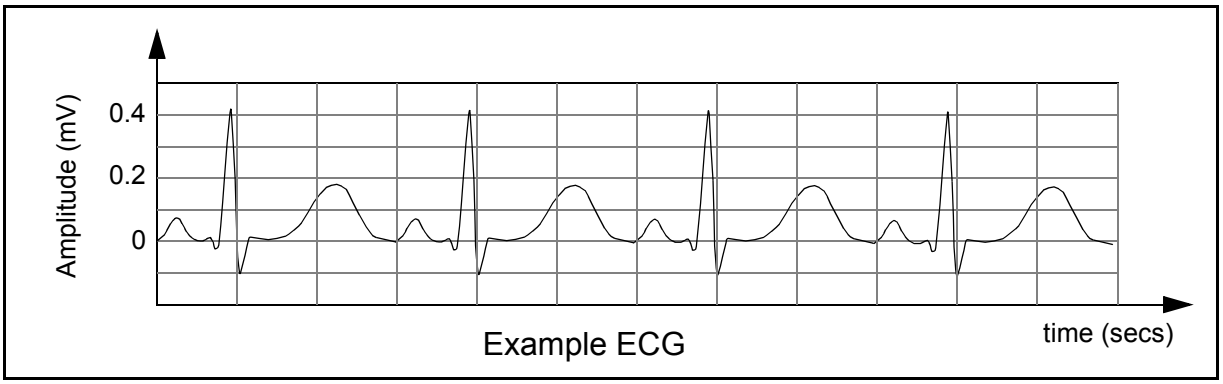
Eigenanalysis: See *Matrix Decompositions - Eigenanalysis*.

Eigenvalue: See *Matrix Decompositions - Eigenanalysis*.

Eigenvector: See *Matrix Decompositions - Eigenanalysis*.

Eight to Fourteen Modulation (EFM): EFM is used in compact disc (CD) players to convert 8 bit symbols to a 14 bit word using a look-up table [33]. When the 14 bit words are used fewer 1-0 and 0-1 transitions are needed than would be the case with the 8 bit words. In addition, the presence of the transitions are guaranteed. This allows required synchronization information to be placed on the disc for every possible data set. In addition, the forced presence of zeros allows the transitions (ones) to occur less frequently than would otherwise be the case. This increases the playing time since more bits can be put on a disk with a fixed minimum feature size (i.e., pit size). See also *Compact Disc.*

Electrocardiogram (ECG): The general name given to the electrical potentials of the heart sensed by electrodes placed externally on the body (i.e., surface leads) [48]. These potentials can also be sensed by placing electrodes directly on the heart as is done with implantable devices (sometimes referred to as pacemakers). The bandwidth used for a typical clinical ECG signal is about 0.05-100Hz. The peak amplitude of a sensed ECG signal is about 1 mV and for use in a DSP system the ECG will typically require to be amplified by a low noise amplifier with gain of about 1000 or more.



Example ECG

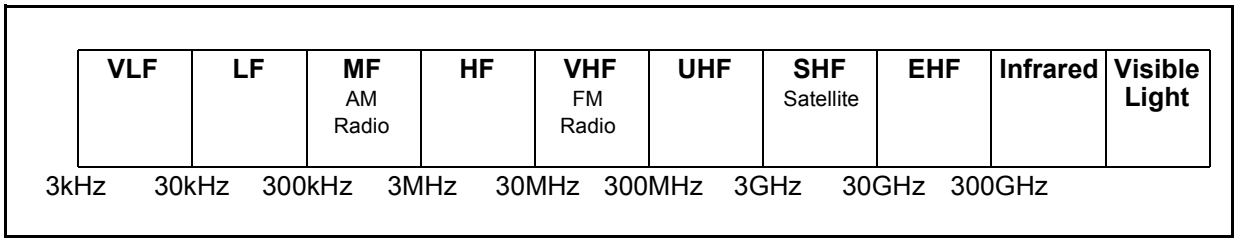
Electroencephalogram (EEG): The EEG measures small microvolt potentials induced by the brain that are picked up by electrodes placed on the head [48]. The frequency range of interest is about 0.5-60Hz. A number of companies are now making multichannel DSP acquisition boards for recording *EEGs* at sampling rates of a few hundred Hertz.

Electromagnetic Interference (EMI): Unwanted electromagnetic radiation resulting from energy sources that interfere with or modulate desired electrical signals within a system.

Electromagnetic Compatibility (EMC): With the proliferation of electronic circuit boards in virtually every walk of life particular care must be taken at the design stage to avoid the electronics

acting as a transmitter of high frequency electromagnetic waves. In general a strip of wire with a high frequency current passing through can act as an antenna and transmit radio waves. The harmonic content from a simple clock in a simple microprocessor system can easily give of radio signals that may interfere with nearby radio communications devices, or other electronic circuitry. A number of EMC regulations have recently been introduced to guard against unwanted radio wave emissions from electronic systems.

Electromagnetic Spectrum: Electromagnetic waves travel through space at approximately 3×10^8 m/s, i.e. the speed of light. In fact, light is a form of electromagnetic radiation for which we have evolved sensors (eyes). The various broadcasting bands are classified as *very low (VLF), low (LF), medium (MF), high (HF), very high (VHF), ultra high (UHF), super high* (SHF), and *extremely high frequencies* (EHF). One of the most familiar bands in everyday life is VHF (very high) used by FM radio stations.



Electromyogram (EMG): Signals sensed by electrodes placed inside muscles of the body. The frequency range of interest is 10-200Hz.

Electroreception: *Electroreception* is a means by which fish, animals and birds use electric fields for navigation or communication. There are two type of electric fish: “strongly electric” such as the electric eel which can uses its electrical energy as a defense mechanism, and; “weakly electric” which applies to many common sea and freshwater fish who use electrical energy for navigation and perhaps even communication [151]. Weakly electric fish can have one of two differing patterns of electric discharge: (1) Continuous wave where a tone like signal is output at frequencies of between 50 and 1000 Hz, and (2) Pulse wave where trains of pulses lasting about a millisecond and spaced about 25 milliseconds apart. The signals are generated by a special tubular organ that extends almost from the fish head to tail. By sensing the variation in electrical conductivity caused by objects distorting the electric field, an electrical image of can be conveyed to the fish via receptors on its body. The relatively weak electric field, however, means that fish are in general electrically short sighted and cannot sense distances any more than one or two fish lengths away. However this is enough to avoid rocks and other poor electrical conductors which will disperse electrical shadows that the fish can pick up on. See also *Mammals*.

Elementary Signals: A set of *elementary signals* can be defined which have certain properties and can be combined in a linear or non linear fashion with time shifts and periodic extensions to create more complicated signals. *Elementary signals* are useful for the mathematical analysis and description of signals and systems [47]. Although there is no universally agreed list of *elementary signals*, a list of the most basic functions is likely to include:

1. Unit Step;
2. Unit Impulse;
3. Rectangular Pulse;
4. Triangular Pulse
5. Ramp Function;

6. Harmonic Oscillation (sine and cosine waves);
7. Exponential Functions;
8. Complex Exponentials;
9. Mother Wavelets and Scaling Functions;

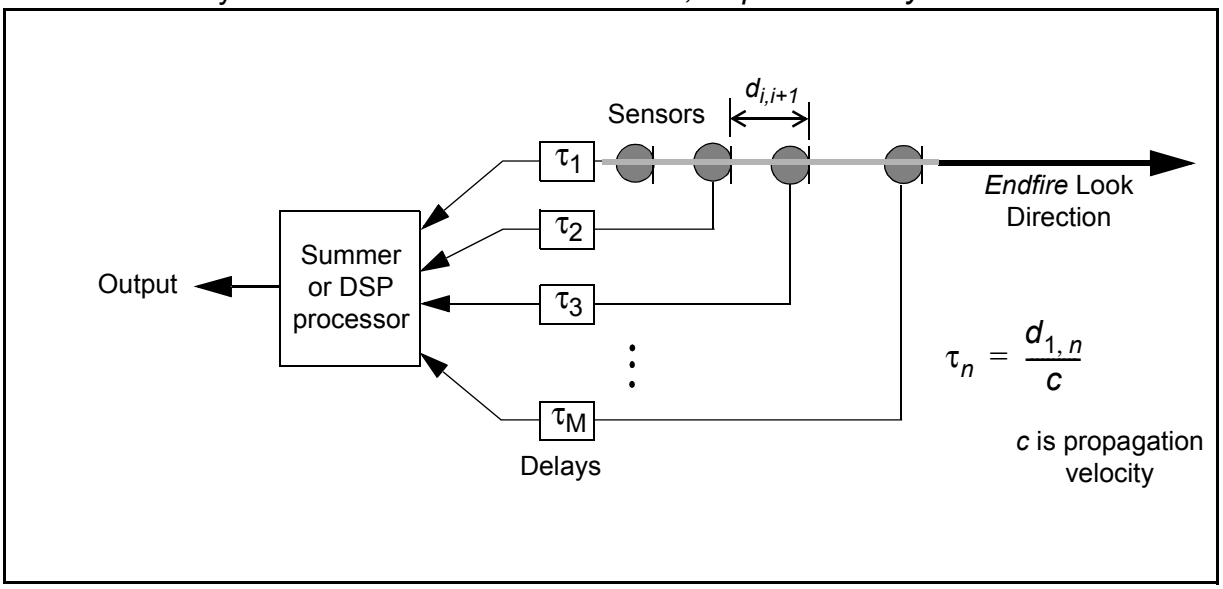
Both analog and discrete versions of the above *elementary signals* can be defined. *Elementary signals* are also referred to as signal primitives. See also *Convolution, Elementary Signals, Fourier Transform Properties, Impulse Response, Sampling Property, Unit Impulse Function, Unit Step Function.*

Elliptic Filter: See *Filters*.

Embedded Control: DSP processors and associated A/D and D/A channels can be used for control of a mechanical system. For example a feedback control algorithm with could be used to control the revolution speed of the blade in a sheet metal cutter. Typically the term *embedded* will imply a real-time system.

Emulator: A hardware board or device which has (hopefully!) the same functionality as an actual DSP chip, and can be used conveniently and effectively for developing and debugging applications before actual implementation on the DSP chip.

Endfire: A beamformer configuration in which the desired signal is located along a line that contains a linear array of sensors. See also *Broadside, Superdirectivity.*



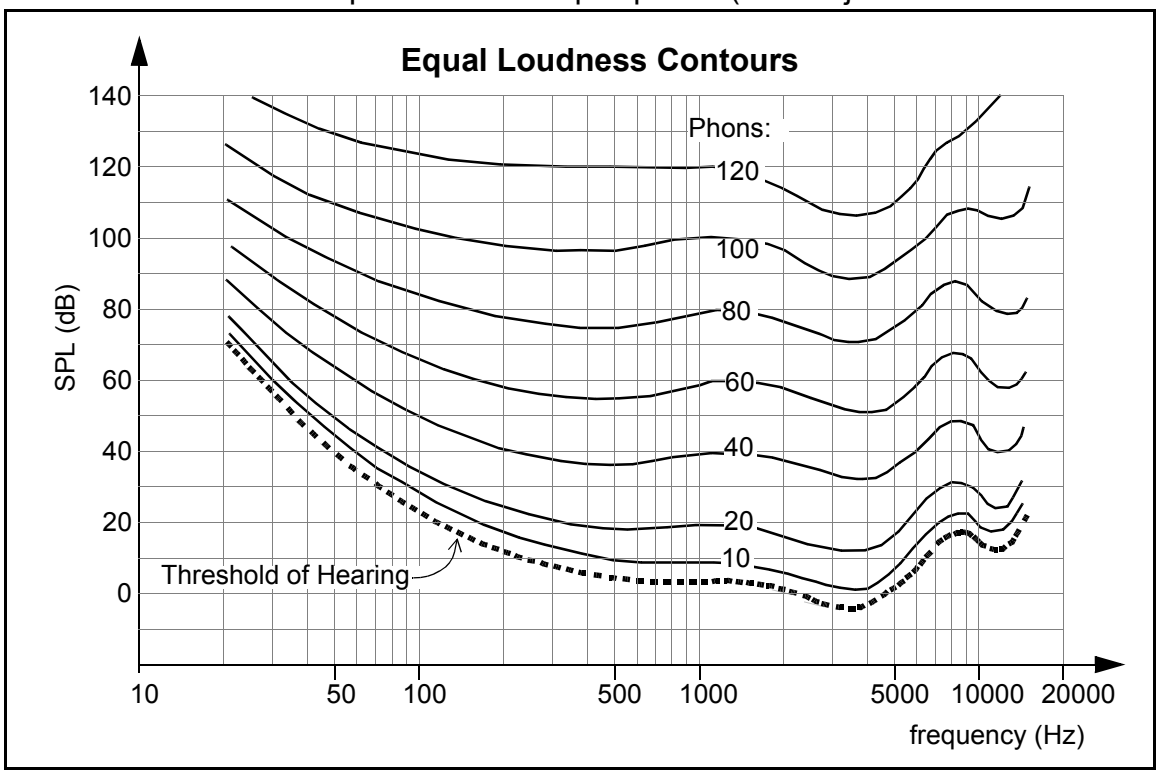
Engaged Tone: See also *Busy Tone*.

Ensemble Averages: A term used interchangeably with statistical average. See *Expected Value.*

Entropy: See *Information Theory*

Entropy Coding: Any type of data compression technique which exploits the fact that some symbols are likely to occur less often than others and assigns fewer bits for coding to the more frequent. For example the letter “e” occurs more often in the English language that the letter “z”. Therefore the transmission code for “e” may only use 2 bits, whereas the transmission code for “z” might require 8 bits. The technique can be further enhanced by assigning codes to comment groups of letters such as “ch”, or “sh”. See also *Huffman Coding*.

Equal Loudness Contours: Equal loudness gives a measure of the actual SPL of a sound compared to the perceived or judged loudness. i.e. a purely subjective measure. The *equal loudness contours* are therefore presented for equal phons (the subjective measure of loudness).

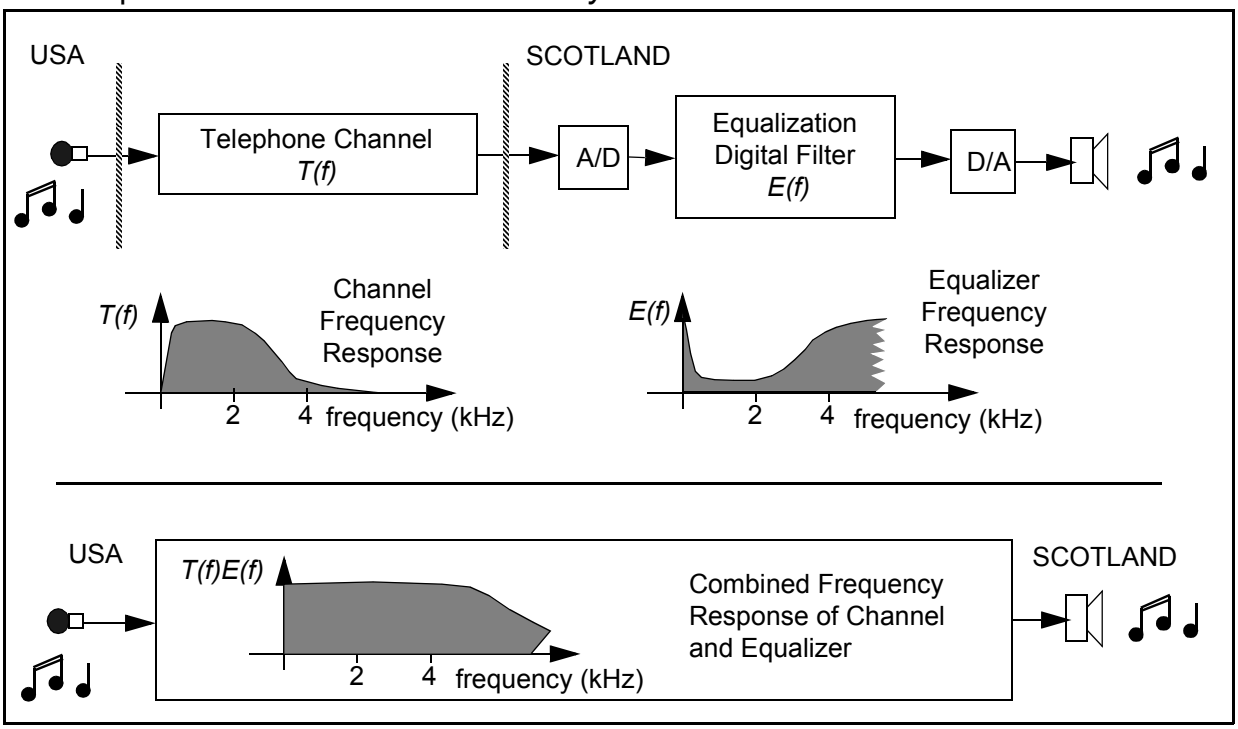


The curves are obtained by averaging over a large cross section of the population who do not have hearing impairments [30]. These measurements were first performed by Fletcher and Munson in 1933 [73], and later by Robinson and Dadson in 1956 [126]. See also *Audiometry, Auditory Filters, Frequency Range of Hearing, Hearing, Loudness Recruitment, Sound Pressure Level, Sound Pressure Level Weighting Curves, Spectral Masking, Temporal Masking, Temporary Threshold Shift, Threshold of Hearing, Ultrasound.*

Equal Tempered Scale: See *Equitempered Scale*.

Equalisation: If a signal is passed through a *channel* (e.g., it is filtered) and the effects of the channel on the signal are removed by making an *inverse* channel filter using DSP, then this is referred to as *equalization*. *Equalization* attempts to restore the frequency and phase characteristic of the signal to the values prior to transmission and is widely used in telecommunications to maximize the reliable transmission data rate, and reduce errors caused by the channel frequency and phase response. *Equalization* implementations are now commonly found in FAX machines and

telephone MODEMS. Most equalization algorithms are adaptive signal processing least squares or least mean squares based. See also *Inverse System Identification*.



Equitempered Scale: Another name for the well known Western music scale of 12 musical notes in an octave where the ratio of the fundamental frequencies of adjacent notes is a constant of value $2^{1/12} = 1.0594631\ldots$. The frequency different between adjacent notes on the *equitempered scale* is therefore about 6%. The difference between the logarithm of the fundamental frequency of adjacent notes is therefore a constant of:

$$\log(2^{1/12}) = 0.0250858\ldots \tag{126}$$

Hence if a piece of digital music is replayed at a sampling rate that mismatches the original by more or less than 6%, the key of the music will be changed (as well as everything sounding that little bit slower!). See also *Music, Music Synthesis, Western Music Scale*.

Equivalent Sound Continuous Level (L_{eq}): Sound pressure level in units of dB (SPL), gives a measure of the instantaneous level of sound. To produce a measure of averaged or integrated sound pressure level a time interval T, the *equivalent sound continuous level* can be calculated [46]:

$$L_{eq,T} = 10\log\left(\frac{\frac{1}{T}\int_0^T P^2(t)}{P^2_{ref}}\right) \tag{127}$$

where P_{ref} is the standard SPL reference pressure of 2×10^{-5} N/m^2 = 20 μPa, and $P(t)$ is the time varying sound pressure. If a particular sound pressure level weighting curve was used, such as the A-weighting scale, then this may be indicated as $L_{Aeq,T}$

L_{eq} measurements can usually be calculated by good quality SPL meters which will average the sound over a specified time typically from a few seconds to a few minutes. SPL meters which provide this facility will correspond to IEC 804: 1985 (and BS 6698 in the UK). See also *Hearing*

Impairment, Sound Exposure Meters, Sound Pressure Level, Sound Pressure Level Weighting Curves, Threshold of Hearing.

Ergodic: If a stationary random process (i.e., a signal) is *ergodic*, then its statistical average (or ensemble average) equal the time average of a single realization of the process. For example given a signal $x(n)$, with a probability density function $p\{x(n)\}$ the mean or expected value is calculated from:

$$\text{Mean of } x(n) = E\{x(n)\} = \sum_n x(n)p\{x(n)\} \tag{128}$$

and the mean squared value is calculated as:

$$\text{Mean Squared Value of } x(n) = E\{[x(n)]^2\} = \sum_n [x(n)]^2 p\{x(n)\} \tag{129}$$

For a stationary signal the probability density function or a number of realizations of the signal may be difficult or inconvenient to obtain. Therefore if the signal is *ergodic* the time averages can be used:

$$E\{x(n)\} \approx \frac{1}{M_2 - M_1} \sum_{n=M_1}^{M_2-1} x(n) \quad \text{for large } (M_2 - M_1) \tag{130}$$

and

$$E\{[x(n)]^2\} \approx \frac{1}{M_2 - M_1} \sum_{n=M_1}^{M_2-1} [x(n)]^2 \quad \text{for large } (M_2 - M_1) \tag{131}$$

See also *Expected Value, Mean Value, Mean Squared Value, Variance, Wide Sense Stationarity.*

Error Analysis: When the cumulative effect of arithmetic round-off errors in an algorithm is calculated, this is referred to as an *error analysis*. Most *error analysis* is performed from consideration of relative and absolute errors of quantities. For example, consider two real numbers x and y, that are estimated as x' and y' with absolute errors Δx and Δy. Therefore:

$$\begin{aligned} x &= x' + \Delta x \\ y &= y' + \Delta y \end{aligned} \tag{132}$$

If x and y are added:

$$w = x + y \tag{133}$$

then the error, Δw, caused by adding the estimated quantities such that $w' = x' + y'$ is calculated by noting that:

$$w = w' + \Delta w = x' + \Delta x + y' + \Delta y \tag{134}$$

and therefore:

$$\Delta w = \Delta x + \Delta y \tag{135}$$

Therefore the (worst case) error caused by the adding (or subtracting) two values is calculated as the sum of the absolute errors.

When the product $z = xy$ is formed then:

$$\begin{aligned} z = xy &= (x' + \Delta x)(y' + \Delta y) \\ &= x'y' + \Delta xy' + \Delta yx' + \Delta x\Delta y \end{aligned} \tag{136}$$

Using the estimated quantities to calculate $z' = x'y'$, the product error, Δz, is given by:

$$\Delta z = z - z' = \Delta xy' + \Delta yx' + \Delta x\Delta y \tag{137}$$

If we assume that the quantities Δx and Δy. are small with respect to x' and y' then the term $\Delta x\Delta y$ can be neglected and the error in the product given by:

$$\Delta z \cong \Delta xy' + \Delta yx' \tag{138}$$

Dividing both sides of the equation by *z*, we can express the relative error in *z* as the sum of the relative errors of *x* and *y*:

$$\frac{\Delta z}{z} \cong \frac{\Delta x}{x} + \frac{\Delta y}{y} \tag{139}$$

The above two results can be used to simplify the *error analysis* of the arithmetic of many signal processing algorithms. See also *Absolute Error, Quantization Noise, Relative Error.*

Error Budget: See *Total Error Budget*.

Error Burst: See *Burst Errors.*

Error Performance Surface: See *Wiener-Hopf Equations*.

Euclidean Distance: Loosely, *Euclidean distance* is simply linear distance, i.e., distance "as the crow flies". More specifically, *Euclidean distance* is the square root of the sum of the squared differences between two vectors. One example would be the distance between the endpoints of the hypotenuse of a right triangle. This distance satisfies the Pythagorean Theorem, i.e., the square root of the sum of the squares. See also *Hamming Distance, Viterbi Algorithm.*

Euler's Formula: An important mathematical relationship in dealing with complex numbers and harmonic relationships is given by Euler's Formula:

$$e^{j\theta} = \cos\theta + j\sin\theta \tag{140}$$

If we think of $e^{j\theta}$ as being a 2-dimensional unit length vector (or phasor) that rotates around the origin as θ is varied, then the real part ($\cos\theta$) is given by the projection of that vector onto the x-axis, and the imaginary part ($\sin\theta$) is given by the projection of that vector onto the y-axis.

European Broadcast Union (EBU): The EBU define standards and recommendations for broadcast of audio, video and data. The EBU has a special relationship with the European Telecommunications Standards Institute (ETSI) through which joint standards are produced such as NICAM 728 (ETS 300 163).

> "a network, in general evolving from a telephony integrated digital network (IDN), that provides end to end connectivity to support a wide range of services including voice and non-voice services, to which users have a limited set of standard multi-purpose user network interfaces."

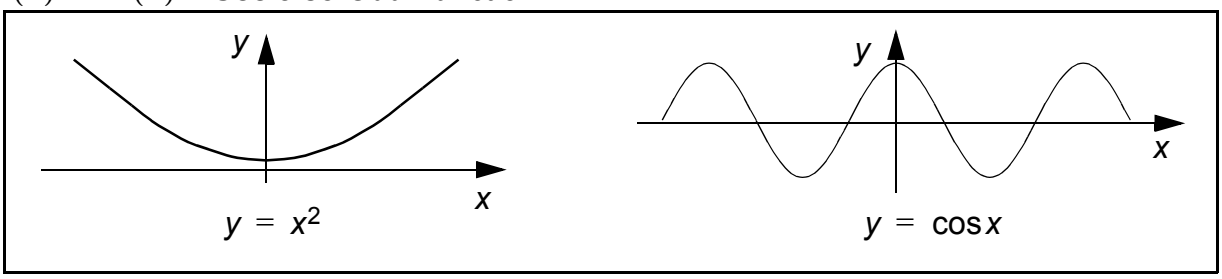
The ITU-T I-series of recommendations fully defines the operation and existence of ISDN. See also *European Telecommunications Standards Institute, International Telecommunication Union, International Organisation for Standards, Standards*, *I-series Recommendations, ITU-T Recommendations*.

European Telecommunications Standards Institute (ETSI): ETSI provides a forum at which all European countries sit to decide upon telecommunications standards. The institute was set up in 1988 for three main reasons: (1) the global (ISO/IEC) standards often left too many questions open; (2) they often do not prescribe enough detail to achieve interoperability; (3) Europe cannot always wait for other countries to agree or follow the standards of the USA and Asia.

ETSI has 12 committees covering telecommunications, wired fixed networks, satellite communications, radio communications for the fixed and mobile services, testing methodology, and equipment engineering. ETSI were responsible for the recommendations of GSM (Group Specialé Mobile, or Global System for Mobile Communications). See also *Comité Européen de Normalisation Electrotechnique, International Telecommunication Union, International Organisation for Standards, Standards*.

Evaluation Board: A printed circuit board produced in volume by a company, and intended for evaluation and benchmarking purposes. An evaluation board is often a cut down version of a production board available from the company. A DSP *evaluation board* is likely to have limited memory available, use a slow clock DSP processor, and be restricted in its convenient expandability. See also *DSP Board*.

Even Function: The graph of an *even function* is symmetric about the y-axis such that $y = f(x) = f(-x)$. This simple 1-dimensional intuition is quickly extended to more complex functions by noting that the basic requirement is still $f(x) = f(-x)$ whether x or $f(x)$ are vectors or vector-valued functions or some combination. Example *even functions* include $y = \cos x$ and $y = x^2$. In contrast an odd function has point symmetry about the origin such that $y = f(x) = -f(x)$. See also *Odd Function*.



Evoked Potentials: When the brain is excited by audio or visual stimuli, small voltage potentials can be measured on the head, emanating from brain [48]. These Visually *Evoked Potentials* (VEP), and Audio *Evoked Potentials* (AEP) can be sampled, and processed using a DSP system. Evoked potentials can also be measured directly on the brain or the brainstem.

Excess Mean Square Error: See *Least Mean Squares (LMS) Algorithm.*

Exp: Common notation used for the exponential function. See *Exponential Function*.

Expected Value: The *expected value,* $E\{.\}$, of a random variable (or a function of a random variable) is simply the average value of the random variable (or of the function of a random variable). The statistical average or mean value of signal $x(n)$ is computed from:

$$\text{Mean of } x(n) = E\{x(n)\} = \sum_n x(n)p\{x(n)\} \tag{141}$$

where $E\{x(n)\}$ is "the *expected value* of $x(n)$", and $p\{x(n)\}$ is the probability density function of the random variable $x(n)$. An another example of *expected values*, the mean squared value of $x(k)$ is calculated as:

$$\text{Mean Squared Value of } x(n) = E\{x^2(n)\} = \sum_n x^2(n)p\{x(n)\} \tag{142}$$

Expected value is a linear operation, i.e.,:

$$E\{ax(n) + by(n)\} = aE\{x(n)\} + bE\{y(n)\} \tag{143}$$

where *a* and *b* are constants and $x(n)$ and $y(n)$ are random signals generated by known probability density functions, $p_y\{y(n)\}$ and $p_x\{x(n)\}$.

For most signals encountered in real time DSP the probability density function is unlikely to be known and therefore the *expected value* cannot be calculated as suggested above. However if the signal is ergodic, then time averages can be used to approximate the statistical averages. See also *Ergodic, Mean Value, Mean Squared Value, Variance, Wide Sense Stationarity*.

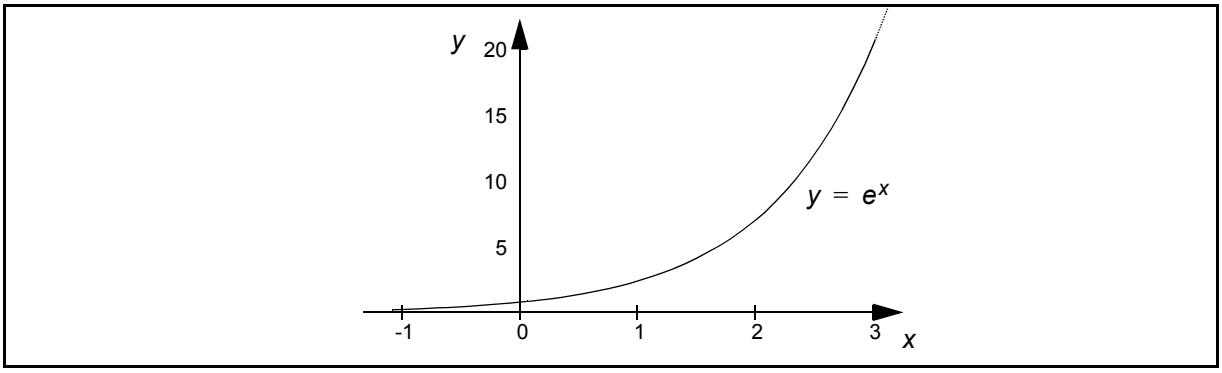
Exponential Averaging: An *exponential averager* with parameter α computes an average $\bar{x}(n)$ of a sequence *{x(n)}* as*:*

$$\bar{x}(n) = (1-\alpha)\bar{x}(n-1) + \alpha x(n) \tag{144}$$

where α is contained in the interval [0,1]. An *exponential average* (a one pole lowpass filter) is simpler to compute than a moving rectangular window since older data points are simply forgotten by the exponentially decreasing powers of (1 - α). A convenient rule of thumb approximation for the "equivalent rectangular window" of an exponential averager is *1/*α data samples. See also *Waveform Averaging, Moving Average, Weighted Moving Average.*

Exponential Function: The simple *exponential function* is:

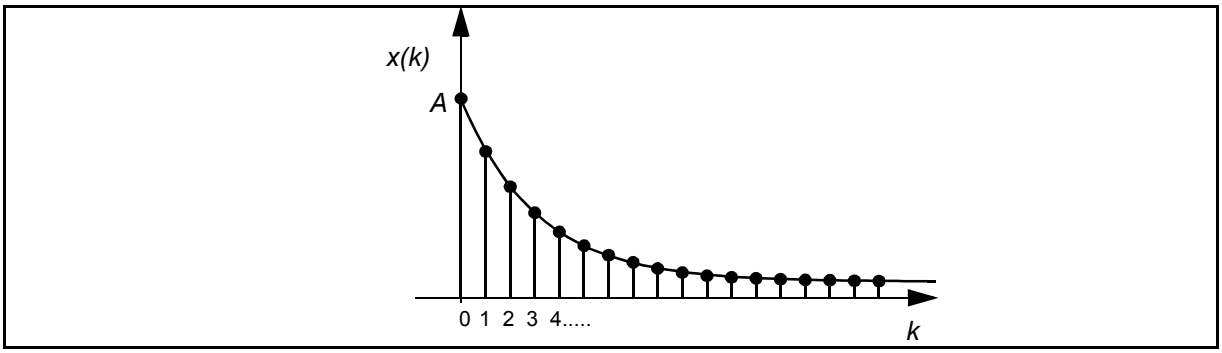
$$y = e^x = \exp(x) \tag{145}$$



where "*e*" is the base of the natural logarithm, $e = 2.7182818$. A key property of the exponential function is that the derivative of e^x is e^x, i.e.

$$\frac{\mathrm{d}}{\mathrm{d}x}e^x = e^x \tag{146}$$

Real causal *exponential functions* can be used to represent the natural decay of energy in a passive system, such as the voltage decay in an RC circuits. For example consider the discrete time exponential:



$$x(k) = Ae^{-\lambda k t_s}u(k) \tag{147}$$

where *u(k)* is the unit step function, t_s is the sampling period, and *A* and λ are constants. See also *Complex Exponential Functions, Damped Sinusoid, RC Circuit.*

F

F-Series Recommendations: The F-series telecommunication recommendations from the International Telecommunication (ITU), advisory committee on telecommunications (denoted ITU-T and formerly known as CCITT) provide standards for services other than telephone (ops, quality, service definitions and human factors). Some of the current recommendations (http://www.itu.ch) include:

F.1 Operational provisions for the international public telegram service.
F.2 Operational provisions for the collection of telegram charges.
F.4 Plain and secret language.
F.10 Character error rate objective for telegraph communication using 5-unit start-stop equipment.
F.11 Continued availability of traditional services.
F.14 General provisions for one-stop-shopping arrangements.
F.15 Evaluating the success of new services.
F.16 Global virtual network service.
F.17 Operational aspects of service telecommunications.
F.18 Guidelines on harmonization of international public bureau services.
F.20 The international gentex service.
F.21 Composition of answer-back codes for the international gentex service.
F.23 Grade of service for long-distance international gentex circuits.
F.24 Average grade of service from country to country in the gentex service.
F.30 Use of various sequences of combinations for special purposes.
F.31 Telegram retransmission system.
F.35 Provisions applying to the operation of an international public automatic message switching service for equipments utilizing the International Telegraph Alphabet No. 2.
F.40 International public telemessage service.
F.41 Interworking between the telemessage service and the international public telegram service.
F.59 General characteristics of the international telex service.
F.60 Operational provisions for the international telex service.
F.61 Operational provisions relating to the chargeable duration of a telex call.
F.63 Additional facilities in the international telex service.
F.64 Determination of the number of international telex circuits required to carry a given volume of traffic.
F.65 Time-to-answer by operators at international telex positions.
F.68 Establishment of the automatic intercontinental telex network.
F.69 The international telex service Service and operational provisions of telex destination codes and telex network identification codes.
F.70 Evaluating the quality of the international telex service.
F.71 Interconnection of private teleprinter networks with the telex network.
F.72 The international telex service - General principles and operational aspects of a store and forward facility.
F.73 Operational principles for communication between terminals of the international telex service and data terminal equipment on packet switched public data networks.
F.74 Intermediate storage devices accessed from the international telex service using single stage selection answerback format.
F.80 Basic requirements for interworking relations between the international telex service and other services.
F.82 Operational provisions to permit interworking between the international telex service and the intex service.
F.86 Interworking between the international telex service and the videotex service.
F.87 Operational principles for the transfer of messages from terminals on the telex network to Group 3 facsimile terminals connected to the public switched telephone network.
F.89 Status enquiry function in the international telex service.
F.91 General statistics for the telegraph services.

F.93 Routing tables for offices connected to the gentex service.
F.95 Table of international telex relations and traffic.
F.96 List of destination indicators.
F.100 Scheduled radiocommunication services.
F.104 International leased circuit services - Customer circuit designations.
F.105 Operational provisions for phototelegrams.
F.106 Operational provisions for private phototelegraph calls.
F.107 Rules for phototelegraph calls established over circuits normally used for telephone traffic.
F.108 Operating rules for international phototelegraph calls to multiple destinations.
F.111 Principles of service for mobile systems.
F.112 Quality objectives for 50-baud start-stop telegraph transmission in the maritime mobile-satellite service.
F.113 Service provisions for aeronautical passenger communications supported by mobile-satellite systems.
F.115 Service objectives and principles for future public land mobile telecommunication systems.
F.120 Ship station identification for VHF/UHF and maritime mobile-satellite services.
F.122 Operational procedures for the maritime satellite data transmission service.
F.125 Numbering plan for access to the mobile-satellite services of INMARSAT from the international telex service.
F.127 Operational procedures for interworking between the international telex service and the service offered by INMARSAT-C system.
F.130 Maritime answer-back codes.
F.131 Radiotelex service codes.
F.140 Point-to-multipoint telecommunication service via satellite.
F.141 International two-way multipoint telecommunication service via satellite.
F.150 Service and operational provisions for the intex service.
F.160 General operational provisions for the international public facsimile services.
F.162 Service and operational requirements of store-and-forward facsimile service.
F.163 Operational requirements of the interconnection of facsimile store-and-forward units.
F.170 Operational provisions for the international public facsimile service between public bureaux (bureaufax).
F.171 Operational provisions relating to the use of store-and-forward switching nodes within the bureaufax service.
F.180 General operational provisions for the international public facsimile service between subscriber stations (telefax).
F.182 Operational provisions for the international public facsimile service between subscribers' stations with Group 3 facsimile machines (Telefax 3).
F.184 Operational provisions for the international public facsimile service between subscriber stations with Group 4 facsimile machines (Telefax 4).
F.190 Operational provisions for the international facsimile service between public bureaux and subscriber stations and vice versa (bureaufax-telefax and vice versa).
F.200 Teletex service.
F.201 Interworking between teletex service and telex service - General principles.
F.202 Interworking between the telex service and the teletex service - General procedures and operational requirements for the international interconnenction of telex/teletex conversion facilities.
F.203 Network based storage for the teletex service.
F.220 Service requirements unique to the processable mode number eleven (PM11) used within teletex service.
F.230 Service requirements unique to the mixed mode (MM) used within the teletex service
F.300 Videotex service.
F.350 Application of T Series recommendations.
F.351 General principles on the presentation of terminal identification to users of the telematic services.
F.353 Provision of telematic and data transmission services on integrated services digital network (ISDN).
F.400 Message handling services: Message Handling System and service overview.
X.400
F.401 Message handling services: naming and addressing for public message handling services.
F.410 Message Handling Services: the public message transfer service.
F.415 Message handling services: Intercommunication with public physical delivery services.

F.420 Message handling services: the public interpersonal messaging service.
F.421 Message handling services: Intercommunication between the IPM service and the telex service.
F.422 Message handling services: Intercommunication between the IPM service and the teletex service.
F.423 Message Handling Services: intercommunication between the interpersonal messaging service and the telefax service.
F.435 Message handling: electronic data interchange messaging service.
F.440 Message handling services: the voice messaging service.
F.500 International public directory services.
F.551 Service for the telematic file transfer within Telefax 3, Telefax 4, Teletex services and message handling services.
F.581 Guidelines for programming communication interfaces (PCIs) definition: Service
F.600 Service and operational principles for public data transmission services.
F.701 Teleconference service.
F.710 General principles for audiographic conference service.
F.711 Audiographic conference teleservice for ISDN.
F.720 Videotelephony services - general.
F.721 Videotelephony teleservice for ISDN.
F.730 Videoconference service- general.
F.732 Broadband Videoconference Services.
F.740 Audiovisual interactive services.
F.761 Service-oriented requirements for telewriting applications.
F.811 Broadband connection-oriented bearer service.
F.812 Broadband connectionless data bearer service.
F.813 Virtual path service for reserved and permanent communications.
F.850 Principles of Universal Personal Telecommunication (UPT).
F.851 Universal personal telecommunication (UPT) - Service description (service set 1)
F.901 Usability evaluation of telecommunication services.
F.902 Interactive services design guidelines.
F.910 Procedures for designing, evaluating and selecting symbols, pictograms and icons.

For additional detail consult the appropriate standard document or contact the ITU. See also *International Telecommunication Union, ITU-T Recommendations, Standards*.

Far End Echo: Signal echo that is produced by components in far end telephone equipment. Far end echo arrives after near end echo. See also *Echo Cancellation, Near End Echo*.

Fast Fourier Transform (FFT): The FFT [66], [93] is a method of computing the discrete Fourier transform (DFT) that exploits the redundancy in the general DFT equation:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}} \quad \text{for } k = 0 \text{ to } N-1 \tag{148}$$

Noting that the DFT computation of Eq. 148 requires approximately N^2 complex multiply accumulates (MACs), where N is a power of 2, the radix-2 FFT requires only $N\log_2 N$ MACs. The computational savings achieved by the FFT is therefore a factor of $N/\log_2 N$. When N is large this

saving can be considerable. The following table compares the number of MACs required for different values of N for the DFT and the FFT:

N	*DFT MACs*	*FFT MACs*
32	1024	160
1024	1048576	10240
32768	~ 1 x 10^9	~ 0.5 x10^6

There are a number of different FFT algorithms sometimes grouped via the names Cooley-Tukey, prime factor, decimation-in-time, decimation-in-frequency, radix-2 and so on. The bottom line for all FFT algorithms is, however, that they remove redundancy from the direct DFT computational algorithm of Eq. 148.

We can highlight the existence of the redundant computation in the DFT by inspecting Eq. 148. First, for notational simplicity we can rewrite Eq. 148 as:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{-kn} \text{ for } k = 0 \text{ to } N-1 \tag{149}$$

where $W = e^{j2\pi/N} = \cos 2\pi/N + j\sin 2\pi/N$ Using the DFT algorithm to calculate the first four components of the DFT of a (trivial) signal with only 8 samples requires the following computations:

$$\begin{aligned}
X(0) &= x(0)+x(1)+x(2)+x(3)+x(4)+x(5)+x(6)+x(7)\\
X(1) &= x(0)+x(1)W_8^{-1}+x(2)W_8^{-2}+x(3)W_8^{-3}+x(4)W_8^{-4}+x(5)W_8^{-5}+x(6)W_8^{-6}+x(7)W_8^{-7}\\
X(2) &= x(0)+x(1)W_8^{-2}+x(2)W_8^{-4}+x(3)W_8^{-6}+x(4)W_8^{-8}+x(5)W_8^{-10}+x(6)W_8^{-12}+x(7)W_8^{-14}\\
X(3) &= x(0)+x(1)W_8^{-3}+x(2)W_8^{-6}+x(3)W_8^{-9}+x(4)W_8^{-12}+x(5)W_8^{-15}+x(6)W_8^{-18}+x(7)W_8^{-21}
\end{aligned} \tag{150}$$

However note that there is redundant (or repeated) arithmetic computation in Eq. 150. For example, consider the third term in the second line of Eq. 150:

$$x(2)W_8^{-2} = x(2)e^{j2\pi\left(\frac{-2}{8}\right)} = x(2)e^{\frac{-j\pi}{2}} \tag{151}$$

Now consider the computation of the third term in the fourth line of Eq. 150:

$$x(2)W_8^{-6} = x(2)e^{j2\pi\left(\frac{-6}{8}\right)} = x(2)e^{\frac{-j3\pi}{2}} = x(2)e^{-j\pi}e^{\frac{-j\pi}{2}} = -x(2)e^{\frac{-j\pi}{2}} \tag{152}$$

Therefore we can save one multiply operation by noting that the term $x(2)W_8^{-6} = -x(2)W_8^{-2}$. In fact because of the periodicity of W_N^{kn} every term in the fourth line of Eq. 150 is available from the computed terms in the second line of the equation. Hence a considerable saving in multiplicative computations can be achieved if the computational order of the DFT algorithm is carefully considered.

More generally we can show that the terms in the second line of Eq. 150 are:

$$x(n)W_8^{-n} = x(n)e^{\frac{-j2\pi n}{8}} = x(n)e^{\frac{-j\pi n}{4}} \tag{153}$$

and for terms in the fourth line of Eq. 150:

$$\begin{aligned} x(n)W_8^{-3n} &= x(n)e^{\frac{-j6\pi n}{8}} = x(n)e^{\frac{-j3\pi n}{4}} = x(n)e^{-j\left(\frac{\pi}{2}+\frac{\pi}{4}\right)n} = x(n)e^{-j\frac{\pi n}{2}}e^{-j\frac{\pi n}{4}} \\ &= x(n)(-j)^n e^{-j\frac{\pi n}{4}} \\ &= (-j)^n x(n)W_8^{-n} \end{aligned} \tag{154}$$

This exploitation of the computational redundancy is the basis of the FFT which allows the same result as the DFT to be computed, but with less MACs.

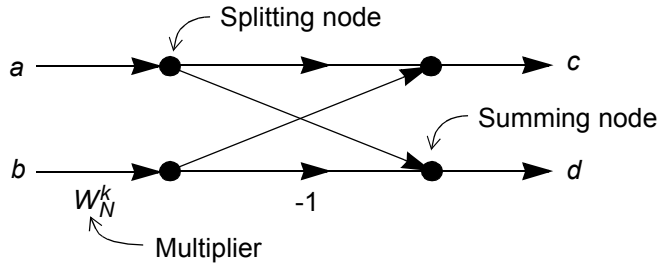
To more formally derive one version of the FFT (decimation-in-time radix-2), consider splitting the DFT equation into two “half signals” consisting of the odd numbered and even numbered samples, where the total number of samples is a power of 2 ($N = 2^n$):

$$\begin{aligned} X(k) &= \sum_{n=0}^{N/2-1} x(2n)e^{\frac{-j2\pi k(2n)}{N}} + \sum_{n=0}^{N/2-1} x(2n+1)e^{\frac{-j2\pi k(2n+1)}{N}} \\ &= \sum_{n=0}^{N/2-1} x(2n)W_N^{-2nk} + \sum_{n=0}^{N/2-1} x(2n+1)W_N^{-(2n+1)k} \\ &= \sum_{n=0}^{N/2-1} x(2n)W_N^{-2nk} + W_N^{-k}\sum_{n=0}^{N/2-1} x(2n+1)W_N^{-2nk} \end{aligned} \tag{155}$$

Notice in Eq. 155 that the N point DFT which requires N^2 MACs in Eq. 148 is now accomplished by performing two $N/2$ point DFTs requiring a total of $2 \times N^2/4$ MACs which is a computational saving of 50%. Therefore a next logical step is to take the $N/2$ point DFTs and perform as $N/4$ point DFTs, saving 50% computation again, and so on. As the number of points we started with was a power of 2, then we can perform this decimation of the signal a total of N times, and each time reduce the total computation of each stage to that of a “butterfly” operation. If $N = 2^n$ then the computational saving is a factor of:

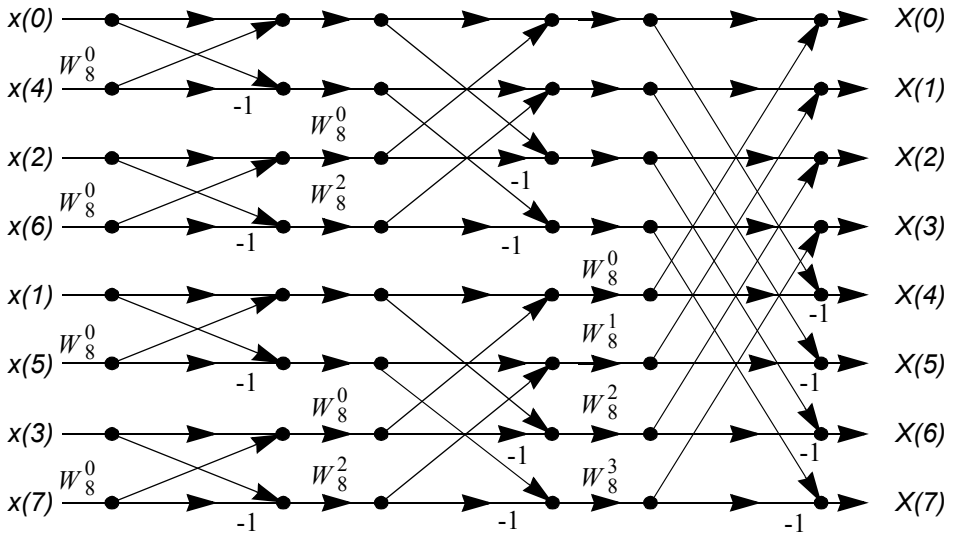
$$\frac{N}{\log_2 N} \tag{156}$$

In general equations for an FFT are awkward to write mathematically, and therefore the algorithm is very often represented as a “butterfly” based signal flow graph (SFG), the butterfly being a simple signal flow graph of the form:



The butterfly signal flow graph. The multipler W_N^k is a complex number, and the input data, *a* and *b* may also be compex. One butterfly computation requires one complex multiply and two complex additions (assuming the data is complex).

A more complete SFG for an 8 point decimation in time radix 2 FFT computation is:



A radix-2 Decimation-in-time (DIT) Cooley-Tukey FFT, for $N = 8$; $W_N^{kn} = e^{-2\pi/N}$. Note that the butterfly computation is repeated through the SFG.

See also Bit Reverse Addressing, Cooley-Tukey, Discrete Cosine Transform, Discrete Fourier Transform, Fast Fourier Transform - Decimation-in-Time (DIT), Fast Fourier Transform - Decimation-in-Frequency (DIF), Fast Fourier Transform - Zero Padding, Fourier, Fourier Analysis, Fourier Series, Fourier Transform, Frequency Response, Phase Response.

Fast Fourier Transform, Decimation-in-Frequency (DIF): The DFT can be reformulated to give the FFT either as a DIT or a DIF algorithm. Since the input data and output data values of the FFT appear in bit-reversed order, *decimation-in-frequency* computation of the FFT provides the output frequency samples in bit-reversed order. See also Discrete Fourier Transform, Fast Fourier Transform, Fast Fourier Transform - Decimation-in-Frequency.

Fast Fourier Transform, Decimation-in-Time (DIT): The DFT can be reformulated to give the FFT either as a DIF or a DIT algorithm. Since the input data and output data values of the FFT appear in bit-reversed order, *decimation-in-time* computation of the FFT provides the output frequency samples in proper order when the input time samples are arranged in bit-reversed order. See also Discrete Fourier Transform, Fast Fourier Transform - Decimation-in-Time, Fast Fourier Transform - Decimation-in-Frequency.

See also Discrete Fourier Transform.

Fast Fourier Transform, Zero Padding: When performing an FFT, the number of data points used in the algorithm is a power of 2 (for radix-2 FFT algorithms). What if a particular process only produces 100 samples and the FFT is required? There are two choices: (1) Truncate the sequence to 64 samples; (2) Pad out the signal by setting the last 28 values of the FFT to be the same as the first 28 samples; (3) Zero pad the data by setting the last 28 values of the FFT to zero.

Solution (1) will lose signal information and solution (2) will add information which is not necessarily part of the signal (i.e. discontinuities). However, solution (3) will only increase the frequency resolution of the FFT by adding more harmonics and does not affect the integrity of the data.

Fast Given's Rotations: See *Matrix Decompositions - Square Root Free Given's Rotations*.

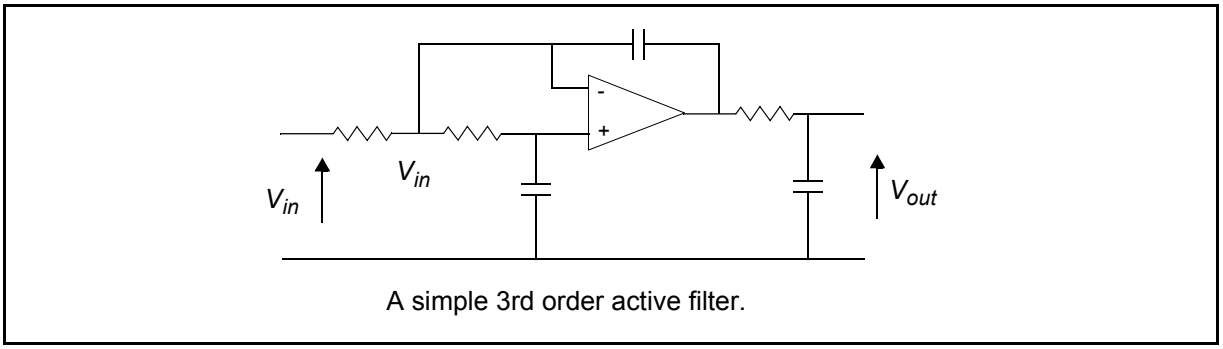
Filtered-U LMS: See *Active Noise Cancellation.*

Filtered-X LMS: See *Least Mean Squares Filtered-X Algorithm.*

Filters: A circuit designed to pass signals of certain frequencies, and attenuate others Filters can be analog or digital [45]. In general a filter with N poles (where N is usually the number of reactive circuit elements used, such as capacitors or inductors) will have a roll-off of $6N$ dB/octave or $20N$ dB/decade.

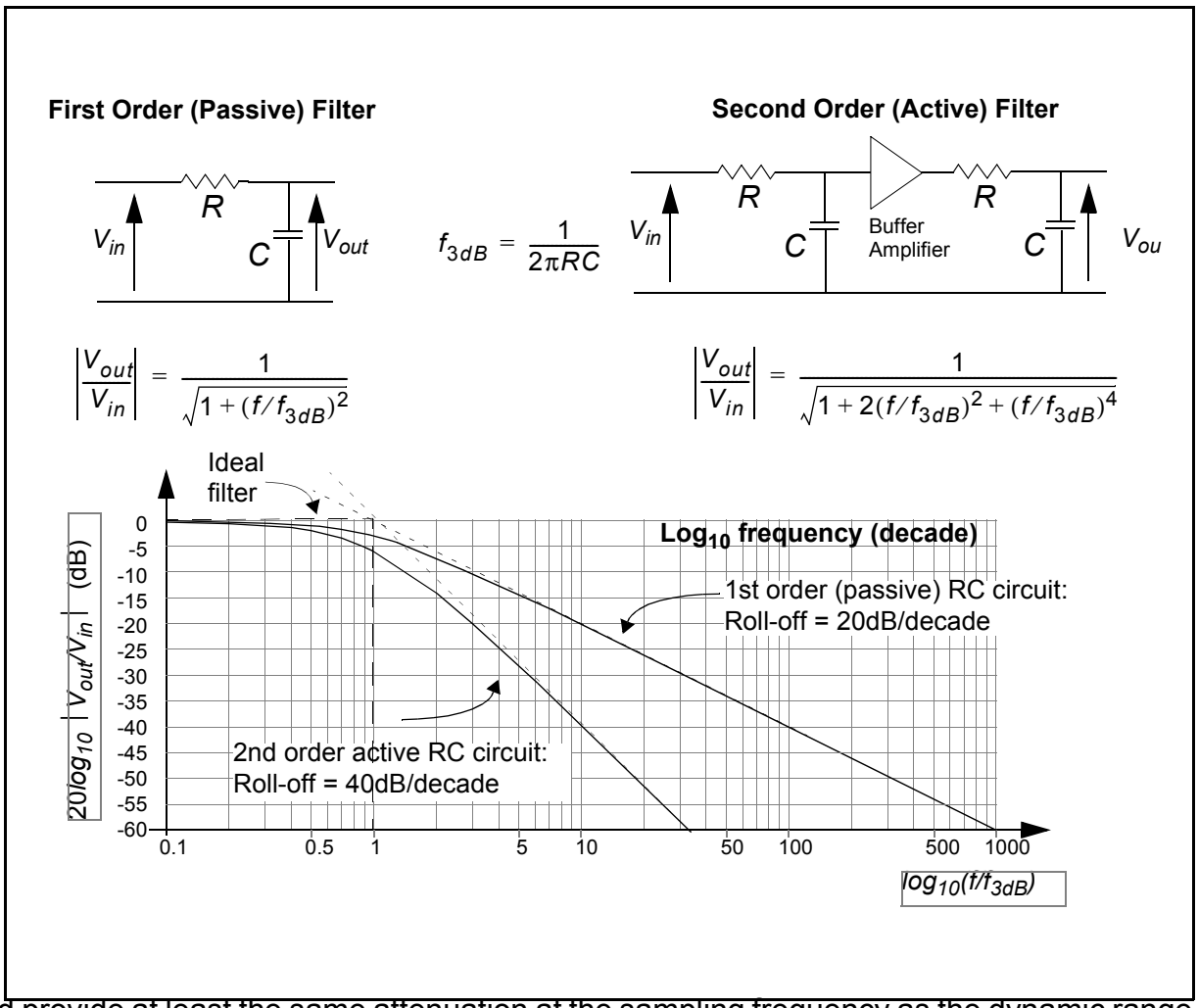
Although the above second order (two pole) *active filter* increases the final rate of roll-off, the sharpness of the knee (at the 3dB frequency) of the filter is not improved and the further increase in order will not produce a filter that approaches the ideal filter. Other designs, such as the Butterworth, Chebychev and Bessel filter, produce filters that have a flatter passband characteristic or a much sharper knee. In general, for a fixed order filter, the sharper the knee of the filter the more variation in the gain of the passband.

A simple active filter is illustrated below.



A simple 3rd order active filter.

The cut-off frequency can be changed by modifying the resistor values. This filter has a roll-off of 18dB/octave, therefore meaning that if used as an anti-alias filter cutting of at $f_s/2$ where f_s is the sampling frequency, the filter would only provide attenuation of 18 dB at f_s and hence aliasing problems may occur. A popular (though not necessarily appropriate) rule of thumb anti-alias filters



should provide at least the same attenuation at the sampling frequency as the dynamic range of the wordlength. For example, if using 16 bit arithmetic the dynamic range is $20\log 2^{16} = 96\text{dB}$ and the roll-off of the filter above the 3dB frequency is at least 96dB/octave. In designing anti-alias fitters, the key requirement is limiting the significance of any aliased frequency components. Because it is the nature of lowpass filters to provide more attenuation at higher frequencies that at lower ones, the aliased components at $f_s/2$ are usually the limiting factor. See also *Active Filter, Anti-alias Filter, Bandpass Filter*, *Digital Filter, High Pass Filter, Low Pass Filter, Knee, Reconstruction Filter, RC Filter, Roll-off.*

Bessel Filter: A filter that has a maximally flat phase response in its passband.

Butterworth Filter: This is a filter based on certain mathematical constraints and defining equations. These filters have been used for a very long time in designing stable analog filters. In general the *Butterworth* filter has a passband that is very flat, at the expense of a slow roll off. The gain of the order *n* (analog) *Butterworth* can be given as

$$\frac{V_{out}}{V_{in}} = \frac{1}{\sqrt{1+(f/f_{3db})^{2n}}} \qquad (157)$$

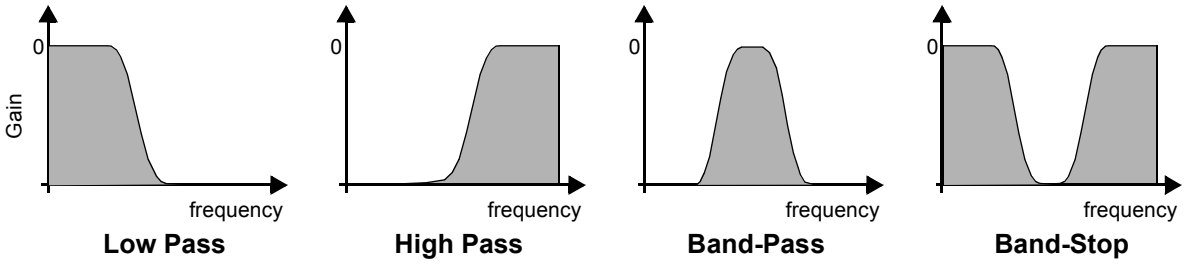
Chebyshev Filter: A type of filter that has a certain amount of ripple in the passband, but has a very steep roll-off. The gain of the order *n* (analog) *Chebyshev* filter can be given as below where C_n is a special

polynomial and ε is a constant that determines the magnitude of the passband ripple. The spelling of *Chebyshev* has many variants (such as Tschebyscheff).

$$\frac{V_{out}}{V_{in}} = \frac{1}{\sqrt{1+\varepsilon^2 C_n^2(f/f_{3db})}} \tag{158}$$

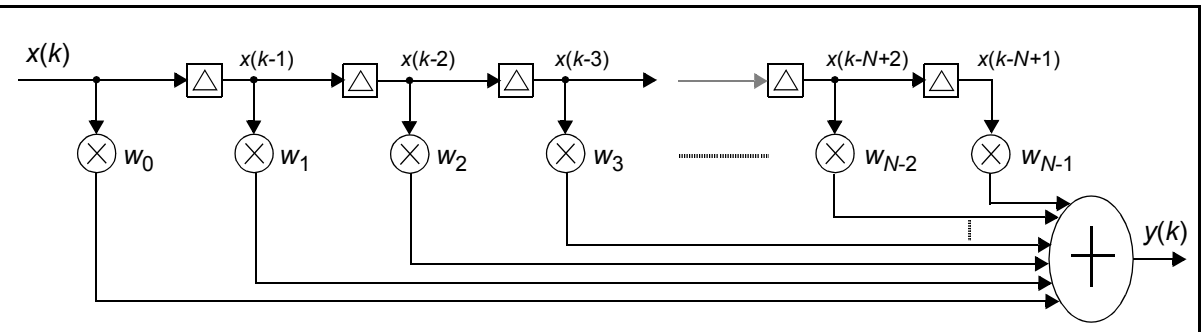
Elliptic Filter: A type of filter that achieves the maximum possible roll-off for a particular filter order. The phase response of an *elliptic filter* is extremely non-linear.

Finite Impulse Response (FIR) Filter: (See first *Digital Filter*). An *FIR filter* digital filter performs a moving weighted average on an input stream of digital data to filter a signal according to some predefined frequency criteria such as a low pass, high pass, band pass, or band-stop filter:



FIR Filters are usually designed with software to be low pass, high pass, band pass or band-stop.

As discussed under *Digital Filter*, an FIR filter is integrated to the real world via analogue to digital converters (ADC) and digital to analogue converters (DAC) and suitable anti-alias and reconstruction filters. An FIR digital filter can be conveniently represented in a signal flow graph:

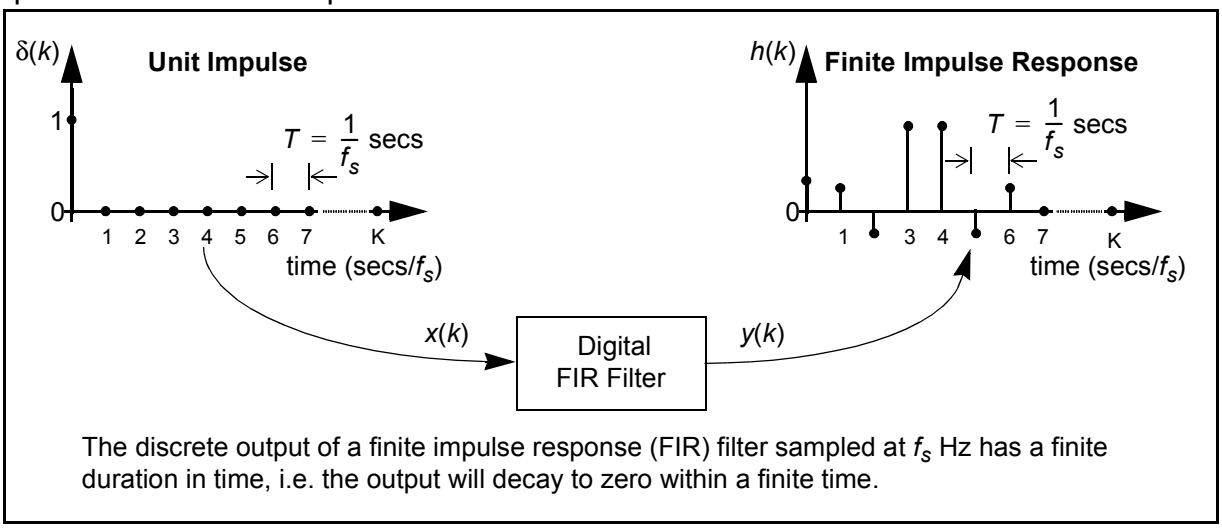


The signal flow graph and the output equation for an FIR *digital filter*. The last N input samples are weighted by the filter coefficients to produce the output $y(k)$

The general output equation (convolution) for an FIR filter is:

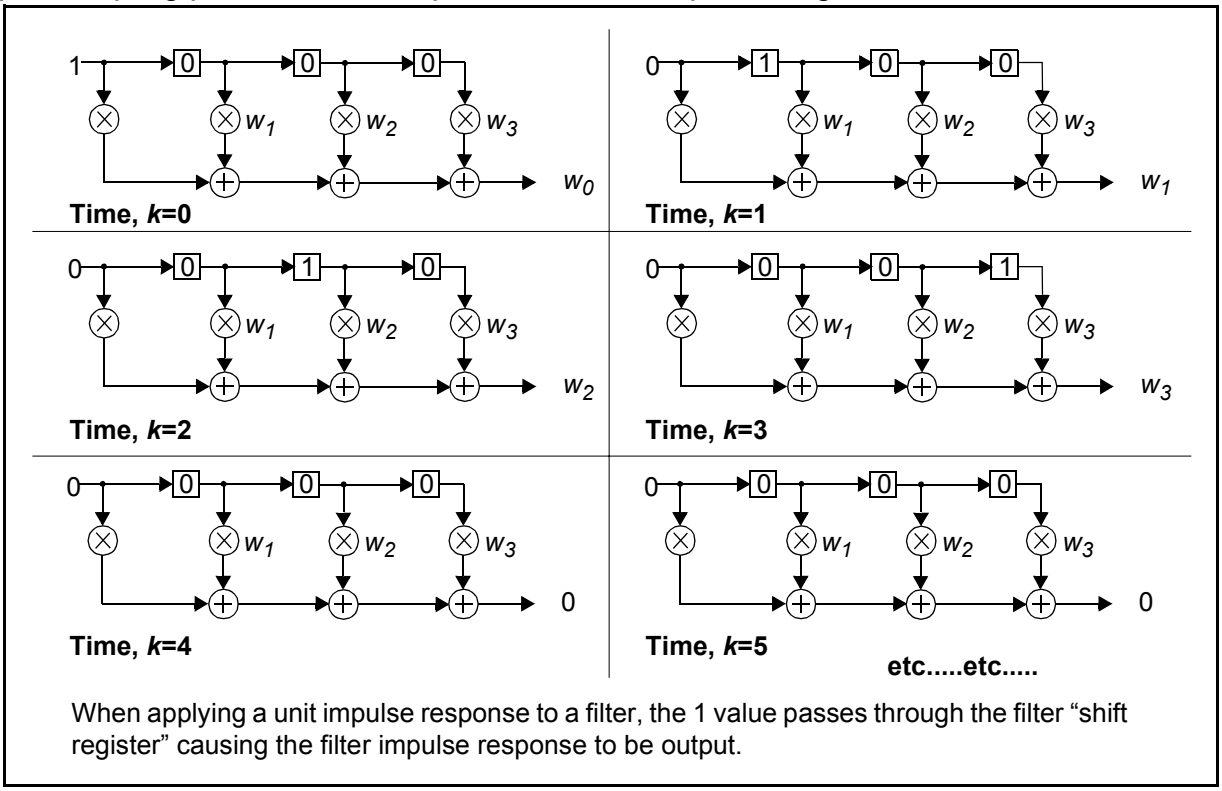
$$\begin{aligned} y(k) &= w_0 x(k) + w_1 x(k-1) + w_2 x(k-2) + w_3 x(k-3) + ... + w_{N-1} x(k-N+1) \\ &= \sum_{n=0}^{N-1} w_n x(k-n) \end{aligned} \tag{159}$$

The term finite impulse response refers to the fact that the impulse response results in energy at only a finite number of samples after which the output is zero. Therefore if the input sequence is a unit impulse the FIR filter output will have a finite duration:



The discrete output of a finite impulse response (FIR) filter sampled at f_s Hz has a finite duration in time, i.e. the output will decay to zero within a finite time.

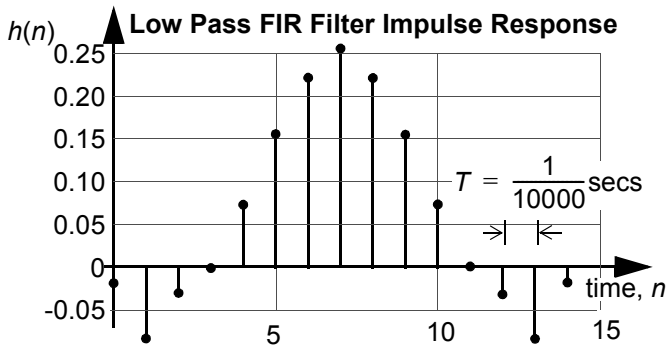
This can be illustrated by considering that the FIR filter is essentially a shift register which is clocked once per sampling period. For example consider a simple 4 weight filter:



When applying a unit impulse response to a filter, the 1 value passes through the filter "shift register" causing the filter impulse response to be output.

As an example, a simple low pass FIR filter can be designed using the DSP design software SystemView by Elanix , with a sampling rate of 10000 Hz, a cut off frequency of around 1000Hz, a

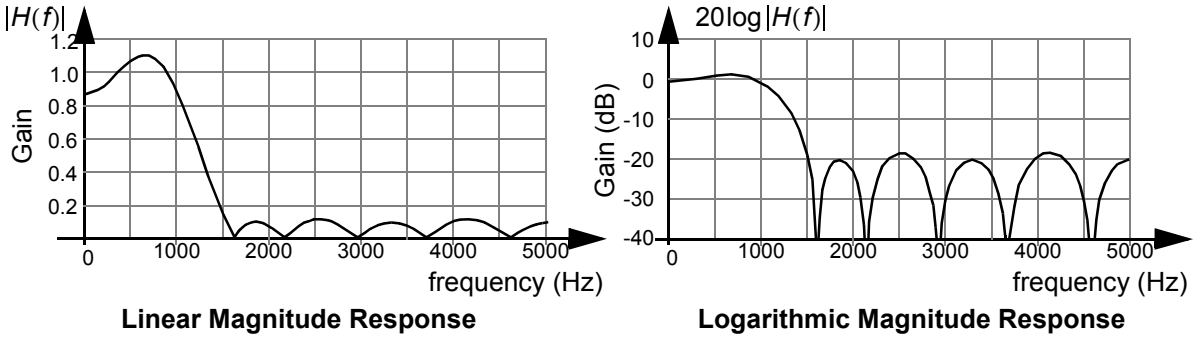
stopband attenuation of about 40dB, passband ripple of less than 1 dB and limited to 15 weights. The resulting filter is:



$w_0 = w_{14}$ = -0.01813...
$w_1 = w_{13}$ = -0.08489...
$w_2 = w_{12}$ =-0.03210...
$w_3 = w_{11}$ = -0.00156...
$w_4 = w_{10}$ = 0.07258...
$w_5 = w_9$ = 0.15493...
$w_6 = w_8$ = 0.22140...
w_7 = 0.25669...
(Truncated to 5 decimal places)

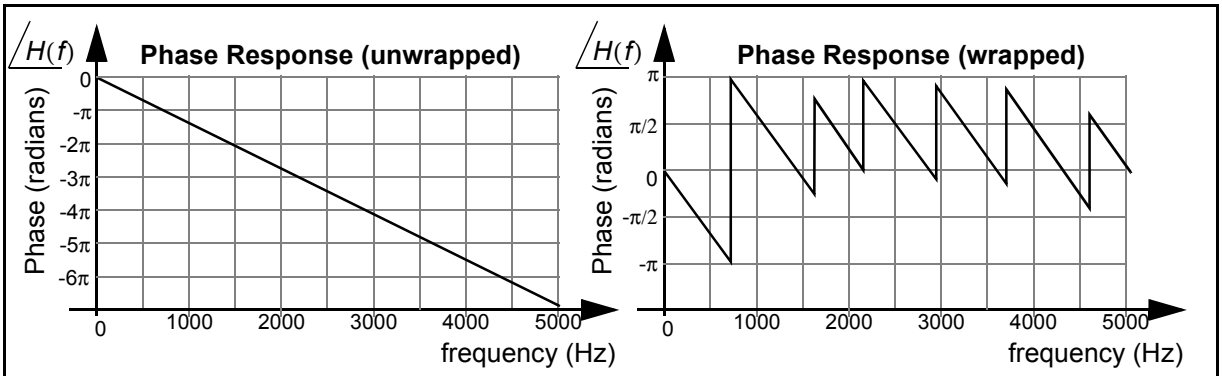
The impulse response $h(n) = w_n$ of a low pass filter, **FIR1** with 15 weights, a sampling rate of 10000 Hz, and cut off frequency designed at around 1000Hz.

Noting that a unit impulse contains “all frequencies”, then the magnitude frequency response and phase response of the filter are found from the DFT (or FFT) of the filter weights:



Linear Magnitude Response **Logarithmic Magnitude Response**

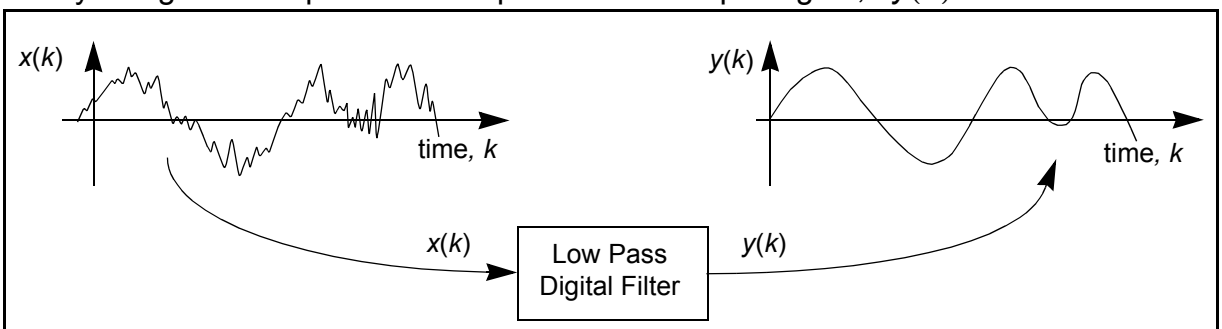
The 1024 point FFT (zero padded) of the above low pass filter impulse response, **FIR1**. As the sampling rate is 10000 Hz the frequency response is only plotted up to 5000 Hz. (Note that the y-axis is labelled Gain rather than Attenuation, this is because -10dB gain is the same as 10dB attenuation. Hence if attenuation was plotted the above figures would be inverted.)



The 1024 point FFT generated phase response (phase shift versus frequency) above low pass filter impulse response, **FIR1**. Note that the the filter is linear phase and the wrapped and unwrapped phase responses are different ways of representing the same information. The "wrapped" phase response will often produced by DSP software packages and gives phase values between $-\pi$ and π only. As the phase is calculated as modulo 2π. i.e. a phase shift of θ is the same as a phase shift of $\theta + 2\pi$ and so on. Phase responses are also often plotted using degrees rather than radians.

From the magnitude and phase response plots we can therefore calculate the attenuation and phase shift of different input signal frequencies. For example, if a single frequency at 1500Hz, with an amplitude of 150 is input to the above filter, then the amplitude of the output signal will be around 30, and phase shifted by a little over -2π radians. However, if a single frequency of 500Hz was input, then the output signal amplitude is amplified by a factor of about 1.085 and phase shifted by about -0.7π radians.

As a more intuitive and illustrative example of filtering, consider inputing the signal, $x(k)$ below to a suitably designed "low pass filter" to produce the output signal, $y(k)$:

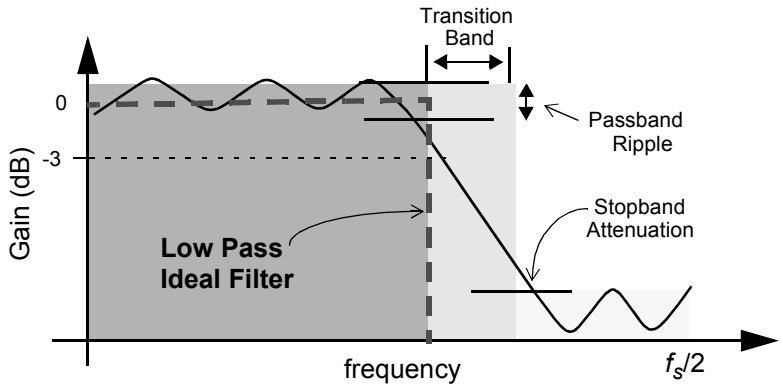


Example of an FIR Filter performing low pass filtering, i.e. removing high frequencies by performing a weighted moving average with suitable low pass characteristic weights. The remaining low frequencies are phase shifted (i.e. time delayed) as a result of passing through the filter.

So, how long is a typical FIR filter? This of course depends on the requirement of the problem being addressed. For the generic filter characteristic shown below more weights are required if:

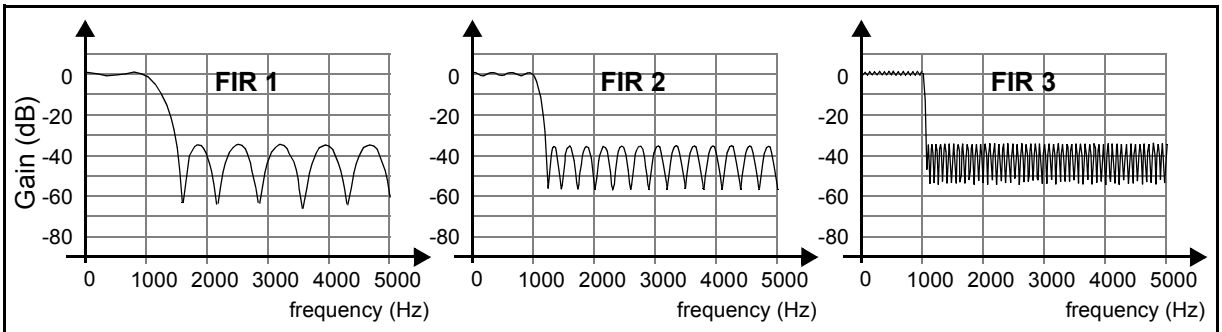
- A sharper transition bandwidth is required;
- More stopband attenuation is required;

- Very small passband ripple is required.



Generic low pass filter magnitude response. The more stringent the filter requirements of stopband attenuation, transition bandwidth and to a lesser extent passband ripple, the more weights that are required.

Consider again the design of the above FIR filter (FIR1) which was a low pass filter cutting of at about 1000Hz. Using SystemView, the above criteria can be varied such that the number of filter weights can be increased and a more stringent filter designed. Consider the design of three low pass filters cutting off at 1000 Hz, with stopband attenuation of 40dB and transition bandwidths 500 Hz, 200 Hz and 50 Hz:



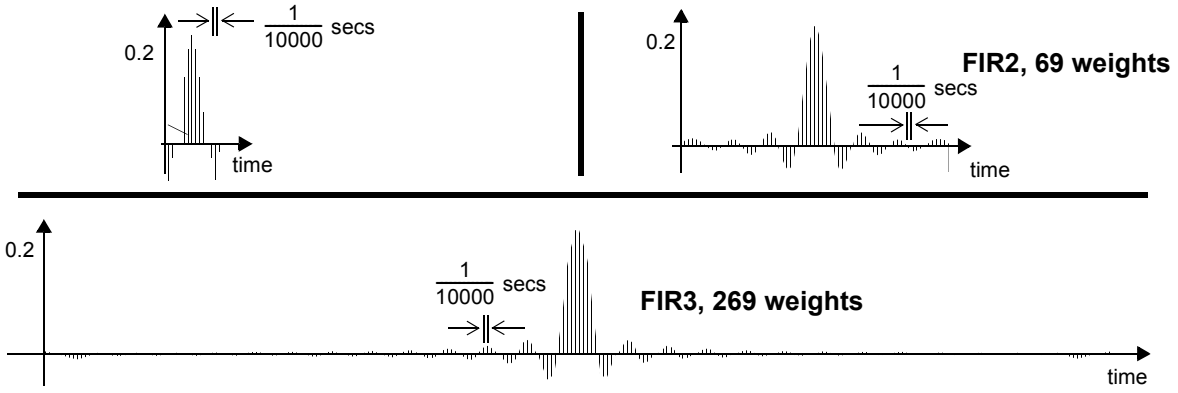
Transition Band: 1000 - 1500Hz
No. of weights: 29

Transition Band: 1000 - 1200Hz
No. of weights: 69

Transition Band: 1000 - 1100Hz
No. of weights: 269

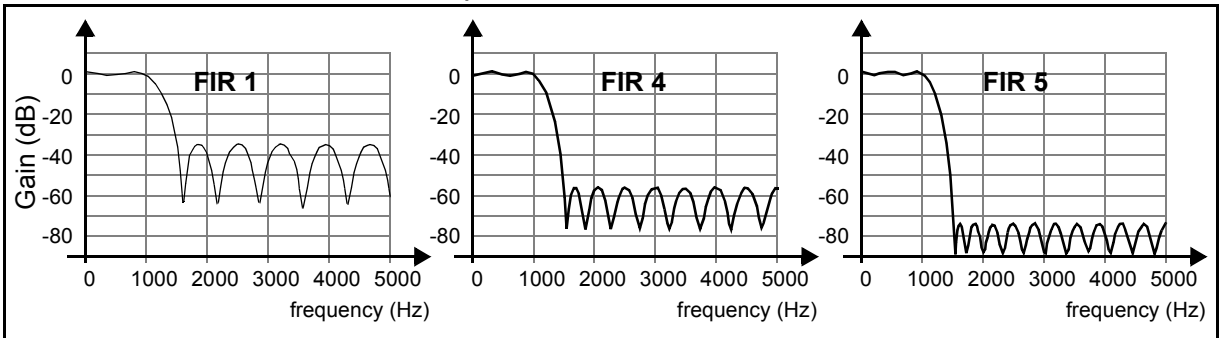
Low pass filters designed parameters: Stopband Attenuation = 40dB; Passband Ripple = 1dB and transition bandwidths, of 500, 200, and 50 Hz. The sharper the transition band the more filter weights that are required.

The respective impulse responses of FIR1 , FIR2 and FIR3 are respectively 15, 69 and 269 weights long, with group delays of 7, 34 and 134 samples respectively.



The impulse responses of low pass filters FIR1, FIR2, and FIR3, all with 40 dB stopband attenuation, 1dB passband ripple, but transition bandwidths of 500, 200 and 50 Hz respectively. Clearly the more stringent the filter parameters, the longer the required impulse response.

Similarly if the stopband attenuation specification is increased, the number of filter weights required will again require to increase. For a low pass filter with a cut off frequency again at 1000 Hz, a transition bandwidth of 500 Hz and stopband attenuations of 40 dB , 60 dB and 80 dB :

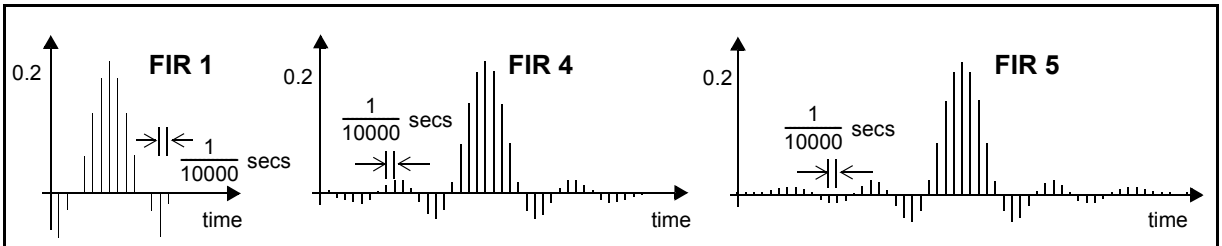


Transition Band: 1000 - 1500Hz
No. of weights: 29

Transition Band: 1000 - 1200Hz
No. of weights: 41

Transition Band: 1000 - 1100Hz
No. of weights: 55

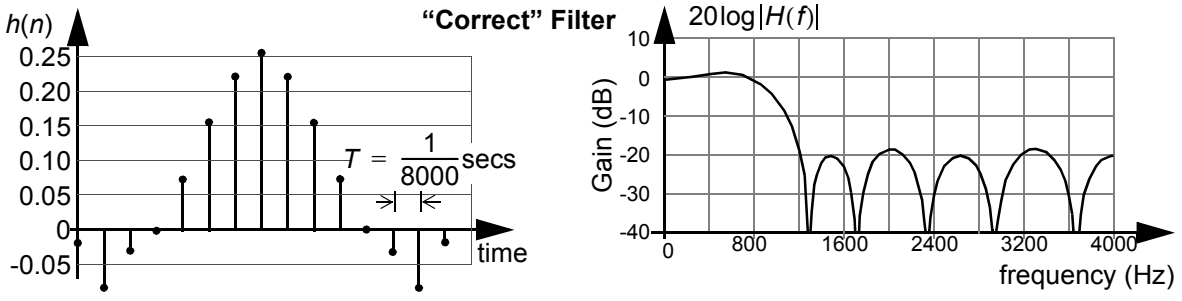
Low pass filters designed parameters: Transition Bandwidth = 500Hz; Passband Ripple = 1dB and stopband attenuations of 40 dB, 60 dB, and 80 dB.



The impulse responses of low pass filters FIR1, FIR4, and FIR5, all 1dB passband ripple, and transition bandwidths of 500 Hz and stopband attenuation of 40, 60 and 80dB respectively. Clearly the more stringent the filter parameters, the longer the required impulse response.

Similarly if the passband ripple parameter is reduced, then a longer impulse response will be required. See also *Adaptive Filter, Digital Filter, Low Pass Filter, High Pass Filter, Bandpass Filter, Bandstop Filter, IIR Filter*.

Finite Impulse Response (FIR) Filter, Bit Errors: If we consider the possibility of a random single bit error in the weights of an FIR filter, the effect on the filter magnitude and phase response can be quite dramatic. Consider a simple 15 weight filter :

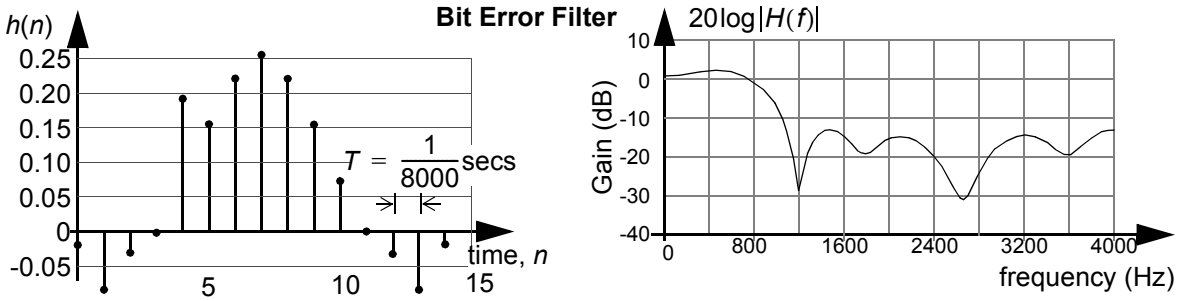


Fifteen weight low pass FIR filter cutting off at 800 Hz.

The 3rd coefficient is of value -0.0725..., and in 16 bit fractional binary notation this is 0.000100101001010_2. If a single bit occurs in the 3rd bit of this binary coefficient then the value becomes:

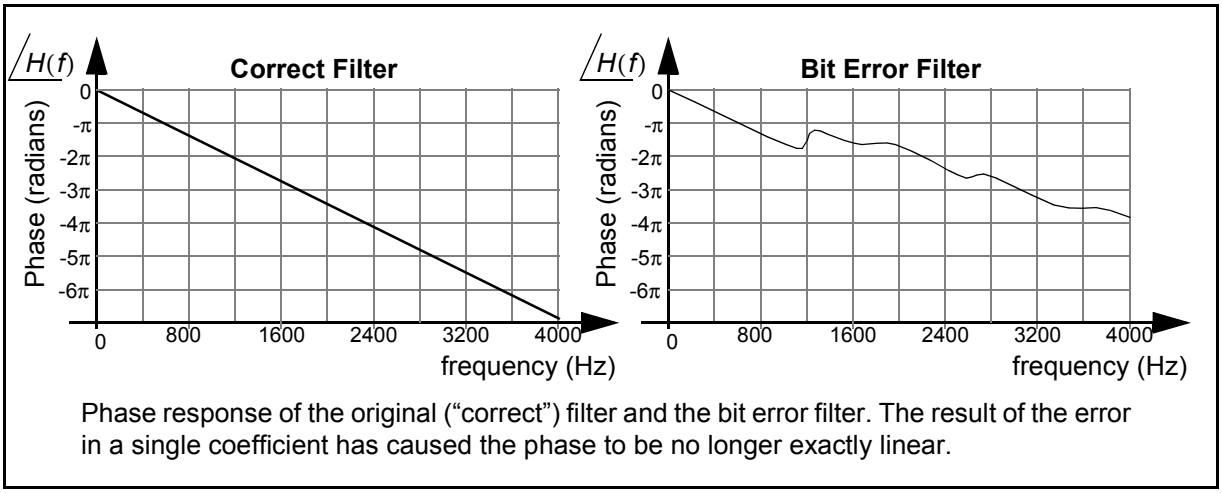
$$0.001100101001010_2 = -0.1957...$$

The impulse response clearly changes "a little" whereas the effect on the frequency response changes is a little more substantial and causes a loss of about 5 dB attenuation.



15 weights low pass FIR filter cutting off at 800 Hz with the 3rd coefficient being in error by a single bit. Note the change to the frequency response compared to the correct filter above.

Also because the impulse response is no longer symmetric the phase response is no longer linear:



Phase response of the original ("correct") filter and the bit error filter. The result of the error in a single coefficient has caused the phase to be no longer exactly linear.

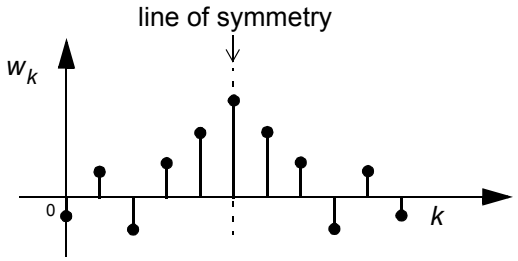
Of course the bit error may have occured at the least significant bits and the frequency domain effect would be much less pronounced. However because of the excellent reliability of DSP processors the occurence of bit errors in filter coefficients is unlikely. See also *Digital Filter, Finite Impulse Response Filter*.

Finite Impulse Response (FIR), Group Delay: See *Finite Impulse Response Filter - Linear Phase*.

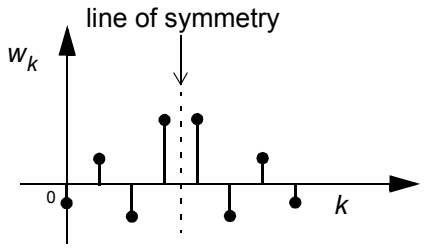
Finite Impulse Response Filter (FIR), Linear Phase: If the weights of an N weight real valued FIR filter are symmetric or anti-symmetric, i.e.

$$: \quad w(n) = \pm w(N-1-n) \tag{160}$$

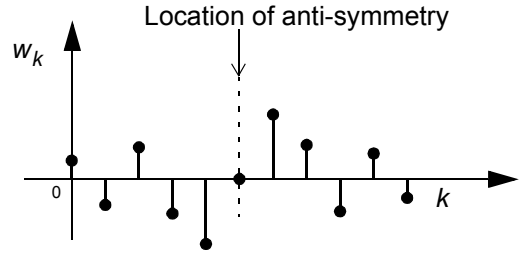
then the filter has linear phase. This means that all frequencies passing through the filter are delayed by the same amount. The impulse response of a linear phase FIR filter can have either an even or odd number of weights.



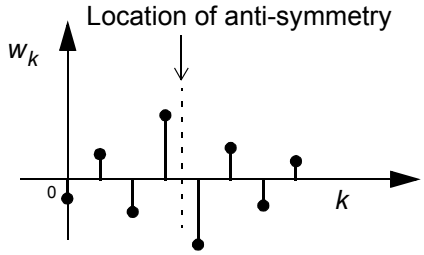
Symmetric impulse response of an 11 (odd number) weight linear phase FIR filter.



Symmetric impulse response of an 8 (even number) weight linear phase FIR filter.



Anti-symmetric impulse response of an 11 (odd number) weight linear phase FIR filter.



Anti-symmetric impulse response of an 8 (even number) weight linear phase FIR filter.

The z-domain plane pole zero plot of a linear phase filter will always have conjugate pair zeroes, i.e. the zeroes are symmetric about the real axis:

The desirable property of linear phase is particularly important in applications where the phase of a signal carries important information. To illustrate the linear phase response, consider inputting a cosine wave of frequency f, sampled at f_s samples per second (i.e. $\cos 2\pi fk/f_s$) to a symmetric impulse response FIR filter with an even number of weights N (i.e. $w_n = w_{N-n}$ for $n = 0, 1, \ldots, N/2-1$). For notational convenience let $\omega = 2\pi f/f_s$:

$$
\begin{aligned}
y(k) &= \sum_{n=0}^{N-1} w_n \cos\omega(k-n) = \sum_{n=0}^{N/2-1} w_n(\cos\omega(k-n) + \cos\omega(k-N+n)) \\
&= \sum_{n=0}^{N/2-1} 2w_n \cos\omega(k-N/2)\cos\omega(n-N/2) \\
&= 2\cos\omega(k-N/2) \sum_{n=0}^{N/2-1} w_n \cos\omega(n-N/2) \\
&= M \cdot \cos\omega(k-N/2), \qquad \text{where } M = \sum_{n=0}^{N/2-1} 2w_n \cos\omega(n-N/2)
\end{aligned}
\tag{161}
$$

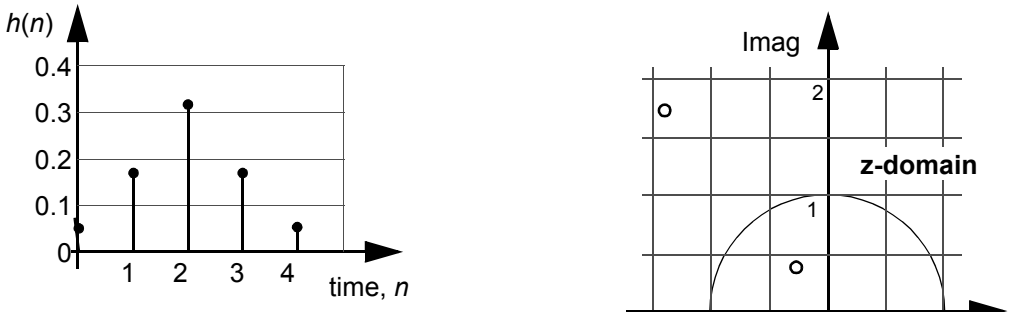
where the trigonometric identity, $\cos A + \cos B = 2\cos((A+B)/2)\cos((A-B)/2)$ has been used. From this equation it can be seen that regardless of the input frequency, the input cosine wave is delayed only by $N/2$ samples, often referred to as the group delay, and its magnitude is scaled by the factor M. Hence the phase response of such an FIR is simply a linear plot of the straight line defined by $\omega N/2$. Group delay is often defined as the differentiation of the phase response with respect to angular frequency. Hence, a filter that provides linear phase has a group delay that is constant for all frequencies. An all-pass filter with constant group delay (i.e., linear phase) produces a pure delay for any input time waveform.

Linear phase FIR filters can be implemented with $N/2$ multiplies and N accumulates compared to the N MACs required by an FIR filter with a non-symmetric impulse response. This can be illustrated by rewriting the output of a symmetric FIR filter with an even number of coefficients:

$$y(k) = \sum_{n=0}^{N-1} w_n x(k-n) = \sum_{n=0}^{N/2-1} w_n [x(k-n) + x(k-N+n)] \quad (162)$$

Although the number of multiplies is halved, most DSP processors can perform a multiply-accumulate in the same time as an addition so there is not necessarily a computational advantage for the implementation of a symmetric FIR filter on a DSP device. One drawback of a linear phase filter is of course that they always introduce a delay.

Linear phase FIR filters are non-minimum phase, i.e. they will always have zeroes that are on are outside of the unit circle. For the z-domain plane plot of the z-transform of a linear phase filter, for all zeroes that are not on the unit circle, there will be a complex conjugate reciprocal of that zero. For example :



The impulse response of a simple 5 weight linear phase FIR filter and the corresponding z-domain plane plot. Note that for the zeroes inside the unit circle at $z = -0.286 \pm 0.3526j$, there are conjugate reciprocal zeroes at:

$$z = \frac{1}{-0.286 \pm 0.3526j} = -1.384 \pm 1.727j$$

See also *Digital Filter, Finite Impulse Response Filter*.

Finite Impulse Response (FIR), Minimum Phase: If the zeroes of an FIR filter all lie within the unit circle on the z-domain plane, then the filter is said to be minimum phase. One simple property is that the inverse filter of a minimum phase FIR filter is a stable IIR filter, i.e. all of the poles lie within the unit circle. See also *Finite Impulse Response Filter.*

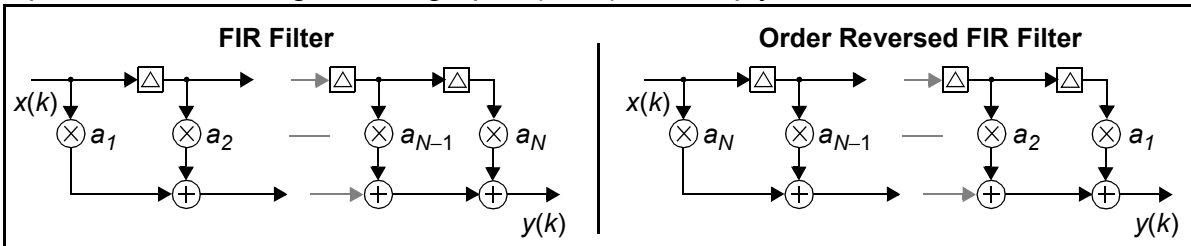
Finite Impulse Response (FIR) Filter, Order Reversed: Consider the general finite impulse response filter with transfer function denoted as $H(z)$:

$$H(z) = a_1 + a_2 z^{-1} + \ldots + a_{N-1} z^{-N+1} + a_N z^{-N} \quad (163)$$

The order reversed FIR filter transfer function, $H_r(z)$ is given by:

$$H_r(z) = a_N + a_{N-1} z^{-1} + \ldots + a_1 z^{-N+1} + a_0 z^{-N} \quad (164)$$

The respective FIR filter signal flow graphs (SFG) are simply:



The signal flow graph for an N+1 weight FIR filter and the order reversed FIR filter. The order reversed FIR filter is same order as the original FIR filter but with the filter weights in opposite order.

From the z-domain functions above it is easy to show that $H_r(z) = z^{-N}H(z^{-1})$. The order reversed FIR filter has exactly the same magnitude frequency response as the original FIR filter:

$$\begin{aligned} \left|H_r(z)\right|_{z=e^{j\omega}} &= \left|z^{-N}H(z^{-1})\right|_{z=e^{j\omega}} = \left|e^{-j\omega N}H(e^{-j\omega})\right| = \left|H(e^{-j\omega})\right| = \left|H(e^{j\omega})\right| \\ &= \left|H(z)\right|_{z=e^{j\omega}} \end{aligned} \tag{165}$$

The phase response of the two filters are however different. The difference to the phase response can be noted by considering that the zeroes of the order reversed FIR filter are the inverse of the zeroes of the original FIR filter, i.e. if the zeroes of Eq. 164 are $\alpha_1, \alpha_2, \ldots \alpha_{N-1}, \alpha_N$:

$$H(z) = (1-\alpha_1 z^{-1})(1-\alpha_2 z^{-1})\ldots(1-\alpha_{N-1}z^{-1})(1-\alpha_N z^{-1}) \tag{166}$$

then the zeroes of the order reversed polynomial are $\alpha_1^{-1}, \alpha_2^{-2}, \ldots \alpha_{N-1}^{-1}, \alpha_N^{-1}$ which can be seen from:

$$\begin{aligned} H_r(z) &= z^{-N}H(z^{-1}) \\ &= z^{-N}(1-\alpha_1 z)(1-\alpha_2 z)\ldots(1-\alpha_{N-1}z)(1-\alpha_N z) \\ &= (z^{-1}-\alpha_1)(z^{-1}-\alpha_2)\ldots(z^{-1}-\alpha_{N-1})(z^{-1}-\alpha_N) \\ &= \frac{(-1)^N}{\alpha_1\alpha_2\ldots\alpha_{N-1}\alpha_N}(1-\alpha_1^{-1}z^{-1})(1-\alpha_2^{-1}z^{-1})\ldots(1-\alpha_{N-1}^{-1}z^{-1})(1-\alpha_N^{-1}z^{-1}) \end{aligned} \tag{167}$$

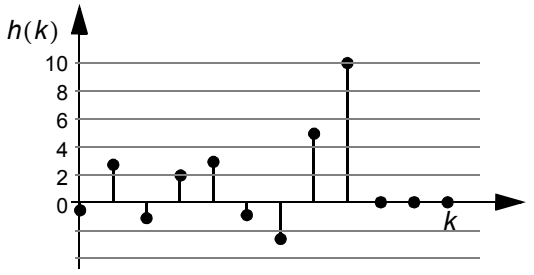
As examples consider the 8 weight FIR filter

$$H(z) = 10 + 5z^{-1} - 3z^{-2} - z^{-3} + 3z^{-4} + 2z^{-5} - z^{-6} + 0.5z^{-7} \tag{168}$$

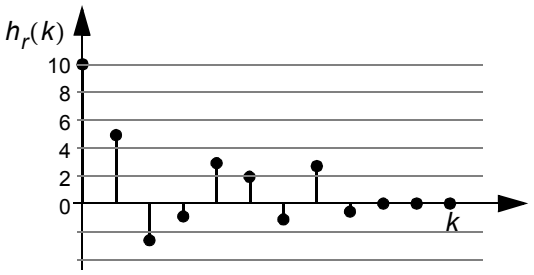
and the corresponding order reversed FIR filter:

$$H(z) = 0.5 - z^{-1} + 2z^{-2} + 3z^{-3} - z^{-4} - 3z^{-5} + 5z^{-6} + 10z^{-7} \tag{169}$$

Assuming a sampling frequency of $f_s = 1$, the impulse response of both filters are easily plotted as :

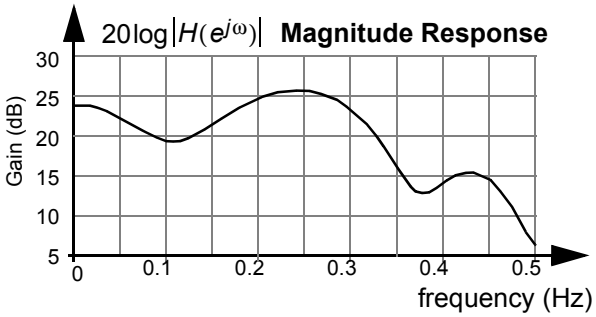


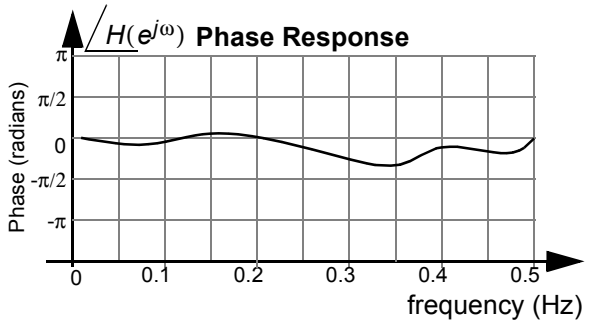
Impulse response, $h(k)$ of simple FIR filter



Order reversed impulse response, $h_r(k)$.

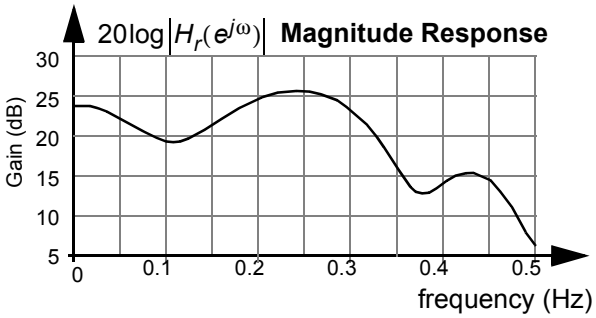
The corresponding magnitude and phase frequency responses of both filters are:

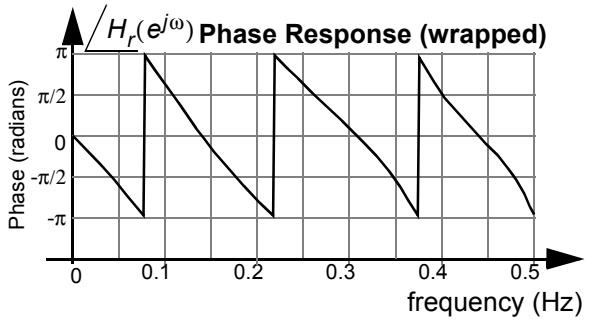




Magnitude and phase frequency response of FIR filter

$$H(z) = 10 + 5z^{-1} - 3z^{-2} - z^{-3} + 3z^{-4} + 2z^{-5} - z^{-6} + 0.5z^{-7}$$





Magnitude and phase frequency response of order reversed FIR filter

$$H_r(z) = 0.5 - z^{-1} + 2z^{-2} + 3z^{-3} - z^{-4} - 3z^{-5} + 5z^{-6} + 10z^{-7}$$

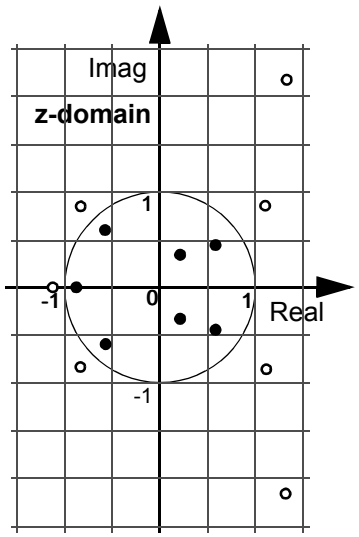
and the z-domain plots of both filter zeroes are:

- **- Zeroes of FIR filter *H*(*z*)**
- **- Zeroes of order reversed FIR filter $H_r(z)$**

For a zero $\alpha = x + jy$ we note that $|\alpha| = \sqrt{x^2 + y^2}$ and therefore for related the order reversed filter zero at $1/\alpha$ we note:

$$\left|\frac{1}{\alpha}\right| = \left|\frac{1}{x+jy}\right| = \frac{|x-jy|}{x^2+y^2} = \frac{x^2+y^2}{\sqrt{x^2+y^2}} = \frac{1}{\sqrt{x^2+y^2}}$$

For this particular example $H(z)$ is clearly minimum phase (all zeroes inside the unit circle), and therefore $H_r(z)$ is maximum phase (all zeroes outside of the unit circle.



See also *All-pass Filter, Digital Filter, Finite Impulse Response Filter*.

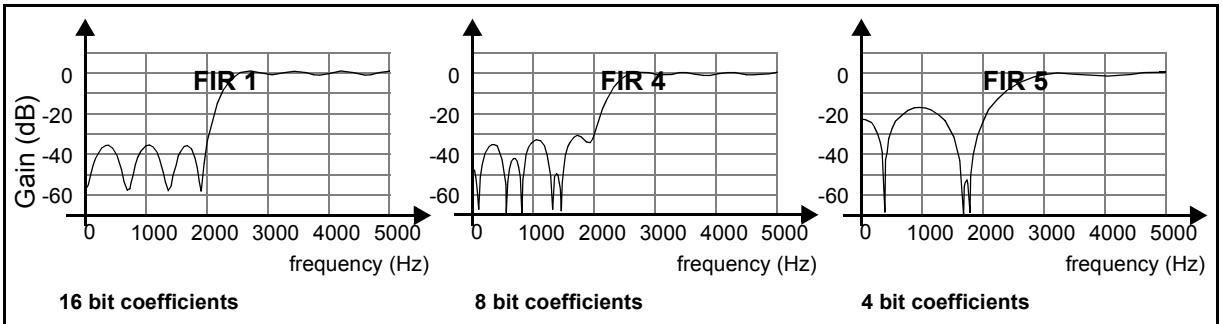
Finite Impulse Response (FIR) Filter, Real Time Implementation: For each input sample, an FIR filter requires to perform *N* multiply accumulate (MAC) operations:

$$y(k) = \sum_{n=0}^{N-1} w_n x(k-n) \tag{170}$$

Therefore if a particular FIR filter is sampling data at f_s Hz, then the number of arithmetic operations per second is:

$$\text{MACs/sec} = Nf_s \tag{171}$$

Finite Impulse Response (FIR) Filter, Wordlength: For a real time implementation of a digital filter, the wordlength used to represent the filter weights will of course have some bearing on the achievable accuracy of the frequency response. Consider for example the design of a high pass digital filter using 16 bit filter weights:

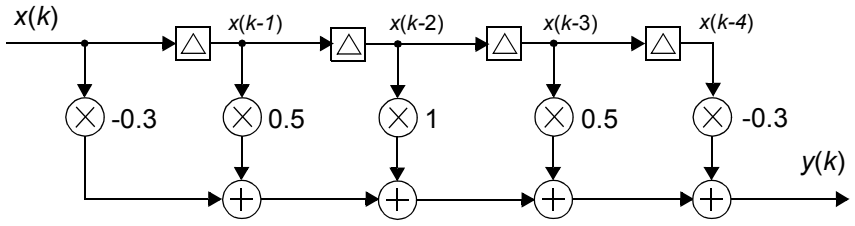


Low pass filters designed parameters: Transition Bandwidth = 500Hz; Passband Ripple = 1dB and stopband attenuations of 40 dB, 60 dB, and 80 dB.

Finite Impulse Response (FIR) Filter, Zeroes: An important way of representing an FIR digital filter is with a z-domain plot of the filter zeroes. By writing the transfer function of an FIR filter in the z-domain, the resulting polynomial in z can be factorised to find the roots, which are in fact the "zeroes" of the digital filter. Consider a simple 5 weight FIR filter :

$$y(k) = -0.3x(k) + 0.5x(k-1) + x(k-2) + 0.5x(k-3) - 0.3x(k-4) \quad (172)$$

The signal flow graph of this filter can be represented as:



The signal flow graph for a 5 weight FIR filter.

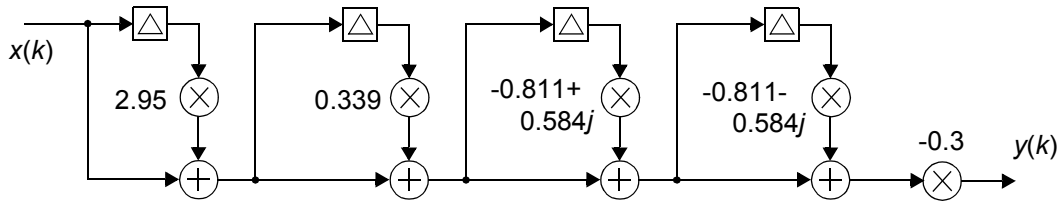
The z-domain transfer function of this polynomial is therefore:

$$H(z) = \frac{Y(z)}{X(z)} = -0.3 + 0.5z^{-1} + z^{-2} + 0.5z^{-3} - 0.3z^{-4} \quad (173)$$

If the z-polynomial of Eq. 173 is factorised (using DSP design software rather than with paper and pencil!) then this gives for this example:

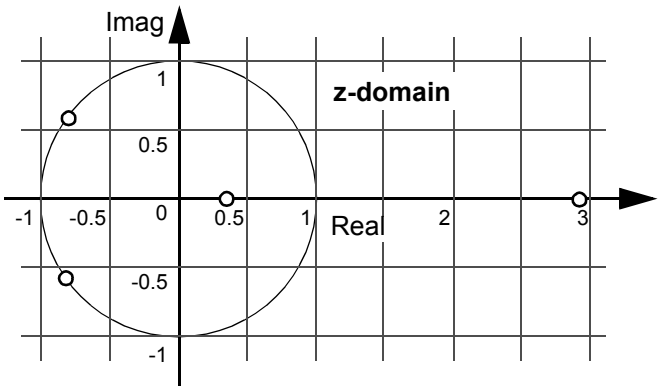
$$H(z) = -0.3(1 - 2.95z^{-1})(1 - (-0.811 + 0.584j)z^{-1})(1 - (-0.811 + 0.584j)z^{-1})(1 - 0.339z^{-1}) \quad (174)$$

and the zeroes of the FIR filter (corresponding to the roots of the polynomial are, $z = 2.95, 0.339, -0.811 + 0.584j$, and $-0.811 - 0.584j$. (Note all quantities have been rounded to 3 decimal places). The corresponding SFG of the FIR filter written in the zero form of Eq. 174 is therefore:



The signal flow graph of four first order cascaded filters corresponding to the same impulse response as the 5 weight filter shown above. The first order filter coefficients correspond to the zeroes of the 5 weight filter.

The zeroes of the FIR filter can also be plotted on the z-domain plane:



The zeroes of the FIR filter in Eq. 173. Note that some of roots are complex. In the case of an FIR filter with real coefficients the zeroes are always symmetric about the x-axis (conjugate pairs) such that when the factorised polynomial is multiplied out there are no imagniary values.

If all of the zeroes of the FIR filter are within the unit circle then the filter is said to be minimum phase.

FIR Filter: See *Finite Impulse Response Filter.*

First Order Hold: Interpolation between discrete samples using a straight line. First order hold is a crude form of interpolation. See also *Interpolation, Step Reconstruction, Zero Order Hold.*

Fixed point: Numbers are represented as integers. 16 bit fixed point can represent a range of 65536 (2^{16}) numbers (including zero). 24 bit fixed point as used by some Motorola fixed point DSP processors can represent a range of 16777216 (2^{24}) numbers. See also *Binary, Binary Point, Floating Point*, *Two's Complement.*

Fixed Point DSP: A DSP processor that can manipulate only fixed point numbers, such as the Motorola DSP56002, the Texas Instruments TMS320C50, the AT&T DSP16, or the Analog Devices ADSP2100. See also *Floating Point DSP.*

Flash Converter: A type (expensive) analog to digital converter.

Fletcher-Munson Curves: Fletcher and Munson's 1933 paper [73] studied the definition of sound intensity, the subjective loudness of human hearing, and associated measurements. Most notably they produced a set of equal loudness contours which showed the variation in SPL of tones at different frequencies that are perceived as having the same loudness. The work of Fletcher and Munson was re-evaluated a few years later by Robinson and Dadson [126]. See also *Equal Loudness Contours, Frequency Range of Hearing, Loudness Recruitment, Sound Pressure Level, Threshold of Hearing*.

Floating Point: Numbers are represented in a *floating point* notation with a mantissa and an exponent. 32 bit floating point numbers have a 24 bit mantissa and an 8 bit exponent. Motorola DSP processors use the IEEE 754 *floating point* number format whereas Texas Instruments use their own *floating point* number format. Both formats give a dynamic range of approximately 2^{-128} to 2^{128} with a resolution of 24 bits.

f_S: Abbreviation for the sampling frequency (in Hz) of a DSP system.

Floating Point Arithmetic Standards: See *IEEE Standard 754.*

Fourier: Jean Baptiste Fourier (died 1830) made a major contribution to modern mathematics with his work in using trigonometric functions to represent heat and diffusion equations. Fourier's work is now collectively refered to as *Fourier Analysis*. See also Discrete Fourier Transform, Fourier Analysis, Fourier Series, Fourier Transform.

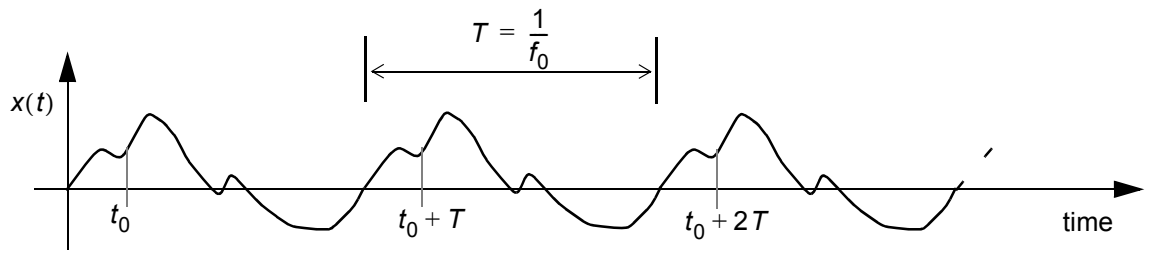
Fourier Analysis: The mathematical tools of the Fourier series, Fourier transform, discrete Fourier transform, magnitude response, phase response and so on can be collectively refered to as *Fourier analysis* tools. Fourier analysis is widely used in science, engineering and business mathematics. In DSP representing a signal in the frequency domain using Fourier techniques, can bring a number of advantages:

> ***Physical Meaning:*** Many real world signals are produced as a sum of harmonic oscillations, e.g. vibrating music strings; vibration induced from the reciprocating motion of an engine; vibration of the vocal tract and other forms of simple harmonic motion. Hence reliable mathematical models can be produced.
>
> ***Filtering:*** It is often useful to filter in a frequency selective manner, e.g. filter out low frequencies.
>
> ***Signal Compression:*** If a signal is periodic over a long time, then rather than transmit the time signal, we can transmit the frequency domain parameters (amplitude, frequencies and phase) and the signal can be reconstructed at the other end of a communications line.

See also Discrete Fourier Transform, Fast Fourier Transform, Fourier Transform.

Fourier Series: There exists mathematical theory called the Fourier series that allows any periodic waveform in time to be decomposed into a sum of harmonically related sine and cosine waves. The first requirement in realising the Fourier series is to calculate the fundamental period, T, which is the shortest time over which the signal repeats, i.e. for a signal $x(t)$, then:

$$x(t) = x(t+T) = x(t+2T) = \ldots = x(t+kT) \tag{175}$$



The (fundamental) period of a signal $x(t)$ identified as T. The fundamental frequency, f_0, is calculated as $f_0 = 1/T$. Clearly $x(t_0) = x(t_0+T) = x(t_0+2T)$.

For a periodic signal with fundamental period T seconds, the *Fourier series* represents this signal as a sum of sine and cosine components that are harmonics of the fundamental frequency, $f_0 = 1/T$ Hz. The Fourier series can be written in a number of different ways:

$$
\begin{aligned}
x(t) &= \sum_{n=0}^{\infty} A_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{\infty} B_n \sin\left(\frac{2\pi nt}{T}\right) \\
&= A_0 + \sum_{n=1}^{\infty} \left[A_n \cos\left(\frac{2\pi nt}{T}\right) + B_n \sin\left(\frac{2\pi nt}{T}\right) \right] \\
&= A_0 + \sum_{n=1}^{\infty} \left[A_n \cos(2\pi n f_0 t) + B_n \sin(2\pi n f_0 t) \right] \\
&= A_0 + \sum_{n=1}^{\infty} \left[A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t) \right] \\
&= \sum_{n=0}^{\infty} \left[A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t) \right] \\
&= A_0 + A_1 \cos(\omega_0 t) + A_2 \cos(2\omega_0 t) + A_2 \cos(3\omega_0 t) + \ldots \\
&\quad + B_1 \sin(\omega_0 t) + B_2 \sin(2\omega_0 t) + B_2 \sin(3\omega_0 t) + \ldots
\end{aligned}
\tag{176}
$$

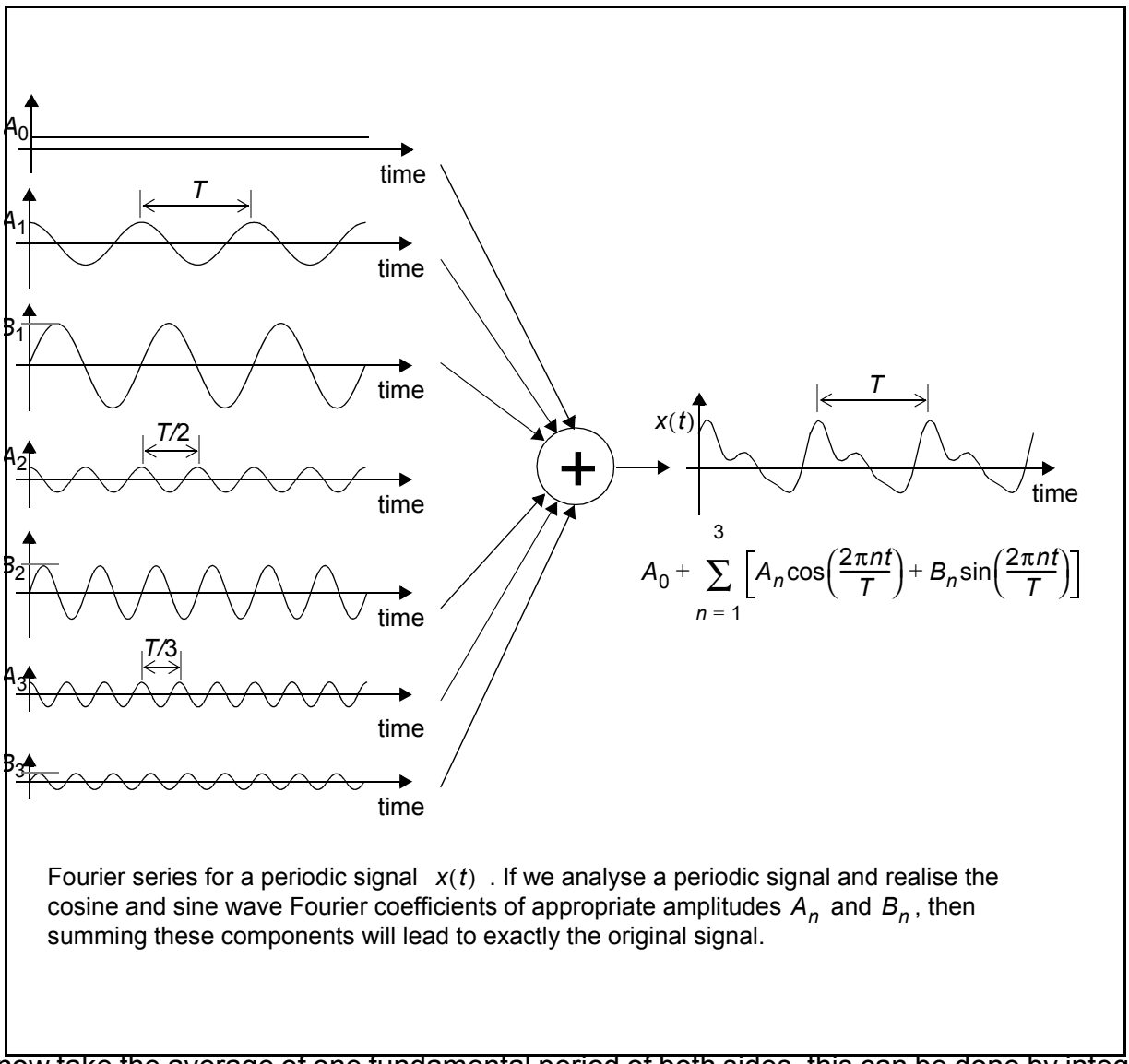
where A_n and B_n are the amplitudes of the various cosine and sine waveforms, and angular frequency is denoted by $\omega_0 = 2\pi f_0$ radians/second.

Depending on the actual problem being solved we can choose to specify the fundamental periodicity of the waveform in terms of the period (T), frequency (f_0), or angular frequency (ω_0) as shown in Eq. 176. Note that there is actually no requirement to specifically include a B_0 term since $\sin 0 = 0$, although there is an A_0 term, since $\cos 0 = 1$, which represents any DC component that may be present in the signal.

In more descriptive language the above Fourier series says that any periodic signal can be reproduced by adding a (possibly infinite) series of harmonically related sinusoidal waveforms of amplitudes A_n or B_n. Therefore if a periodic signal with a fundamental period of say 0.01 seconds is identified, then the Fourier series will allow this waveform to be represented as a sum of various cosine and sine waves at frequencies of 100 Hz (the fundamental frequency, f_0), 200Hz, 300Hz (the harmonic frequencies $2f_0$, $3f_0$) and so on. The amplitudes of these cosine and sine waves are given by A_0, A_1, B_1, A_2, B_2, A_3 ..... and so on.

So how are the values of A_n and B_n calculated? The answer can be derived by some basic trigonometry. Taking the last line of Eq. 176, if we multiply both sides by $\cos(p\omega_0 t)$, where p is an arbitrary positive integer, then we get:

$$
\cos(p\omega_0 t)x(t) = \cos(p\omega_0 t) \sum_{n=0}^{\infty} \left[A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t) \right] \tag{177}
$$



Fourier series for a periodic signal $x(t)$. If we analyse a periodic signal and realise the cosine and sine wave Fourier coefficients of appropriate amplitudes A_n and B_n, then summing these components will lead to exactly the original signal.

If we now take the average of one fundamental period of both sides, this can be done by integrating the functions over any one period, T:

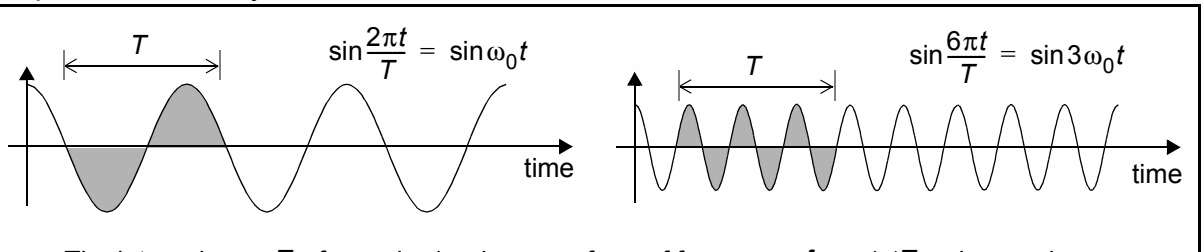
$$\int_0^T \cos(p\omega_0 t)x(t)dt = \int_0^T \left\{ \cos(p\omega_0 t) \sum_{n=0}^{T} A_n \cos(n\omega_0 t) + \sum_{n=0}^{T} B_n \sin(n\omega_0 t) \right\} dt$$

$$= \sum_{n=0}^{\infty} \int_0^T \{A_n \cos(p\omega_0 t)\cos(n\omega_0 t)\}dt + \sum_{n=0}^{\infty} \int_0^T \{B_n \cos(p\omega_0 t)\sin(n\omega_0 t)\}dt \quad (178)$$

Noting the zero value of the second term in the last line of Eq. 178, i.e. :

$$\begin{aligned}\int_0^T \{B_n \cos(p\omega_0 t)\sin(n\omega_0 t)\}dt &= \frac{B_n}{2}\int_0^T (\sin(p+n)\omega_0 t - \sin(p-n)\omega_0 t)dt \\ &= \frac{B_n}{2}\int_0^T \sin\left[\frac{(p+n)2\pi t}{T}\right]dt - \frac{B_n}{2}\int_0^T \sin\left[\frac{(p-n)2\pi t}{T}\right]dt \\ &= 0\end{aligned} \tag{179}$$

using the trigonometric identity $2\cos A \sin B = \sin(A+B) - \sin(A-B)$ and noting that the integral over one period, T, of any harmonic of the term $\sin[2\pi t/T]$ is zero:



The integral over T of any sine/cosine waveform of frequency $f_0 = 1/T$ or harmonics thereof, $2f_0, 2f_0, 3f_0, \ldots$ is zero, regardless of the amplitude or phase of the signal.

Eq. 179 is true for all values of the positive integers p and n.

For the first term in the last line of Eq. 178 the average is only zero if $p \neq n$, i.e. :

$$\int_0^T A_n \cos(p\omega_0 t)\cos(n\omega_0 t)\ dt = \frac{A_n}{2}\int_0^T (\cos(p+n)\omega_0 t + \cos(p-n)\omega_0 t)dt = 0, \quad p \neq n \tag{180}$$

this time using the trigonometric identity $2\cos A \cos B = \cos(A+B) + \cos(A-B)$.

If $p = n$ then:

$$\begin{aligned}\int_0^T A_n \cos(n\omega_0 t)\cos(n\omega_0 t)\ dt &= A_n \int_0^T \cos^2(n\omega_0 t)\ dt \\ &= \frac{A_n}{2}\int_0^T (1 + \cos 2n\omega_0 t)dt = \frac{A_n}{2}\int_0^T 1dt = \frac{A_n t}{2}\Big|_0^T = \frac{A_n T}{2}\end{aligned} \tag{181}$$

Therefore using Eqs. 179, 180, 181 in Eq. 178 we note that:

$$\int_0^T \cos(p\omega_0 t)x(t)dt = \frac{A_n T}{2} \tag{182}$$

and therefore:

$$A_n = \frac{2}{T}\int_0^T x(t)\cos(n\omega_0 t)dt \tag{183}$$

By premultiplying and time averaging Eq. 178 by $\sin(p\omega_0 t)$ and using a similar set of simplifications to Eqs. 179, 180, 181 we can similarly show that:

$$B_n = \frac{2}{T}\int_0^T x(t)\sin(n\omega_0 t)dt \tag{184}$$

Hence the three key equations for calculating the Fourier series of a periodic signal with fundamental period T are:

$$x(t) = \sum_{n=0}^{\infty} A_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{\infty} B_n \sin\left(\frac{2\pi nt}{T}\right)$$

$$A_n = \frac{2}{T}\int_0^T x(t)\cos(n\omega_0 t)dt \tag{185}$$

$$B_n = \frac{2}{T}\int_0^T x(t)\sin(n\omega_0 t)dt$$

Fourier Series Equations

See also Basis Function, Discrete Cosine Transform, Discrete Fourier Transform, Fast Fourier Transform, Fourier, Fourier Analysis, Fourier Series - Amplitude/Phase Representation, Fourier Series - Complex Exponential Representation, Fourier Transform, Frequency Response, Impulse Response, Gibbs Phenomenon, Parseval's Theorem.

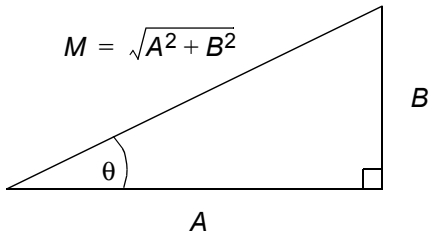
Fourier Series, Amplitude/Phase Representation: It is often useful to abbreviate the notation of the Fourier series such that the series is a sum of cosine (or sine) only terms with a phase shift. To perform this notational simplification, first consider the simple trigonometric function:

$$A\cos\omega t + B\sin\omega t \tag{186}$$

where A and B are real numbers. If we introduce another variable, M such that $M = \sqrt{A^2 + B^2}$ then:

$$
\begin{aligned}
A\cos\omega t + B\sin\omega t &= \frac{\sqrt{A^2+B^2}}{\sqrt{A^2+B^2}}(A\cos\omega t + B\sin\omega t) \\
&= M\left(\frac{A}{\sqrt{A^2+B^2}}\cos\omega t + \frac{B}{\sqrt{A^2+B^2}}\sin\omega t\right) \\
&= M(\cos\theta\cos\omega t + \sin\theta\sin\omega t) \\
&= M\cos(\omega t - \theta) \\
&= \sqrt{A^2+B^2}\cos(\omega t - \{\tan^{-1}B/A\})
\end{aligned}
\tag{187}
$$

since θ is the angle made by a right angle triangle of hypotenuese M and sides of A and B, i.e. $\tan^{-1}(B/A) = \theta$.

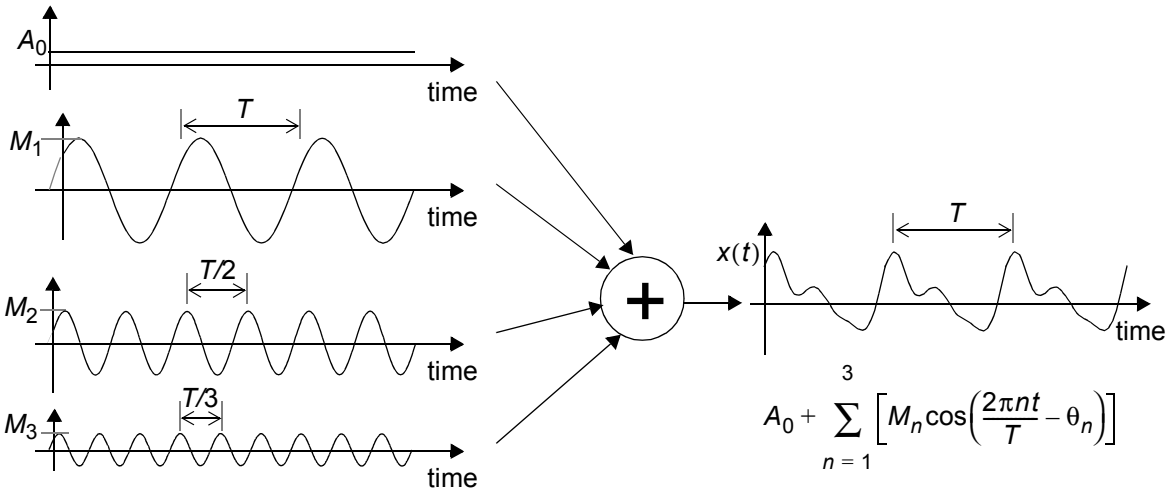


A simple right angled triangle with arbitrary length sides of A and B. The sine of the angle θ is the ratio of the opposite side over the hypotenuese, B/M and the cosine of the angle θ is the ratio of the adjacent side over the hypotenuese, A/M. the tangent of the angle θ is the ratio of the opposite side over the adjacent side, B/A.

This result shows that the sum of a sine and a cosine waveform of arbitrary amplitudes is a sinusoidal signal of the same frequency but different amplitude and phase from the original sine and cosine terms. Using this result of Eq. 187 to combine each sine and cosine term, we can rewrite the Fourier series of Eq. 176 as:

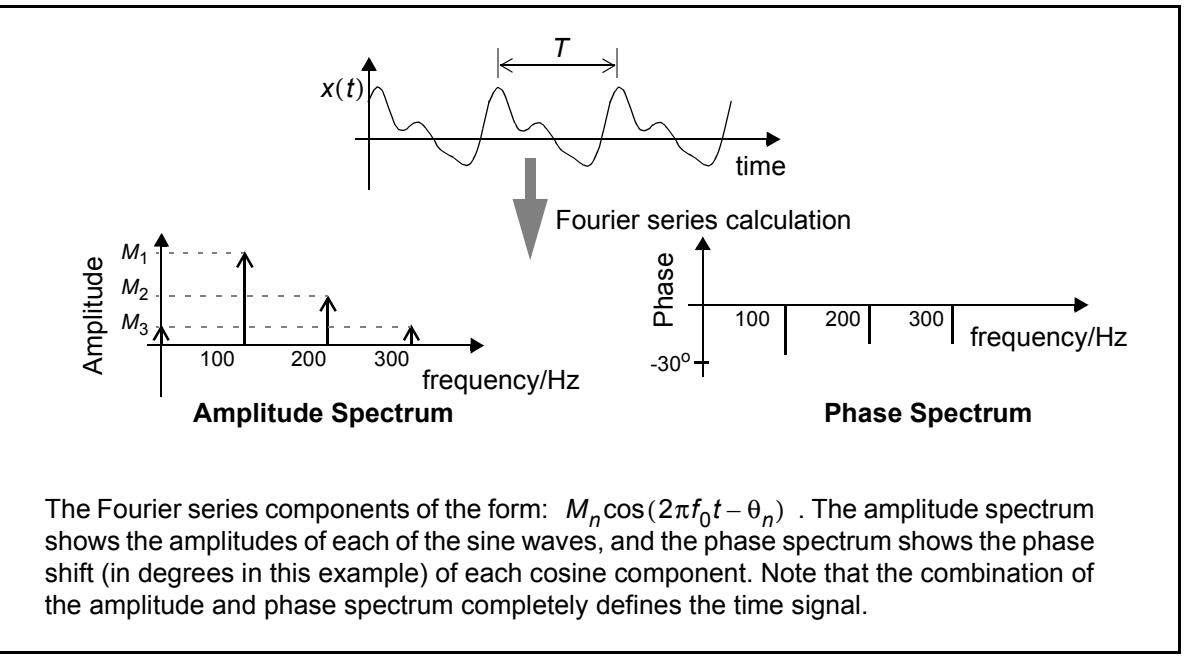
$$
\begin{aligned}
x(t) &= \sum_{n=0}^{\infty} A_n\cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{\infty} B_n\sin\left(\frac{2\pi nt}{T}\right) \\
x(t) &= \sum_{n=0}^{\infty} M_n\cos(n\omega_0 t - \theta_n) \\
\theta_n &= \tan^{-1}B_n/A_n \\
M_n &= \sqrt{A_n^2+B_n^2}
\end{aligned}
\tag{188}
$$

where A_n and B_n are calculated as before using Eqs. 183 and 184.



Comparing this Fourier series with the one on page 156 note that the sine and cosine terms have been combined for each frequency to produce a single cosine waveform of amplitude $M_n = \sqrt{A_n^2 + B_n^2}$ and phase $\theta_n = B/A$.

From this representation of the Fourier series, can plot an *amplitude line spectrum* and a *phase spectrum*:



The Fourier series components of the form: $M_n \cos(2\pi f_0 t - \theta_n)$. The amplitude spectrum shows the amplitudes of each of the sine waves, and the phase spectrum shows the phase shift (in degrees in this example) of each cosine component. Note that the combination of the amplitude and phase spectrum completely defines the time signal.

See also Discrete Cosine Transform, Discrete Fourier Transform, Fast Fourier Transform - Zero Padding, Fourier, Fourier Analysis, Fourier Series, Fourier Series - Complex Exponential Representation, Fourier Transform, Impulse Response, Gibbs Phenomenon, Parseval's Theorem.

Fourier Series, Complex Exponential Representation: It can be useful and instructive to represent the *Fourier series* in terms of complex exponentials rather than sine and cosine waveforms. (In the derivation presented below we will assume that the signal under analysis is real valued, although the result extends easily to complex signals.) From Euler's formula, note that:

$$e^{j\omega} = \cos\omega + j\sin\omega \quad \Rightarrow \cos\omega = \frac{e^{j\omega} + e^{-j\omega}}{2} \quad \text{and} \quad \sin\omega = \frac{e^{j\omega} - e^{-j\omega}}{2j} \tag{189}$$

Substituting the complex exponential definitions for sine and cosine in Eq. 176 (defined in item *Fourier Series*) and rearranging gives:

$$\begin{aligned} x(t) &= A_0 + \sum_{n=1}^{\infty} A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t) \\ &= A_0 + \sum_{n=1}^{\infty} \left[A_n \left(\frac{e^{jn\omega_0 t} + e^{-jn\omega_0 t}}{2} \right) + B_n \left(\frac{e^{jn\omega_0 t} - e^{-jn\omega_0 t}}{2j} \right) \right] \\ &= A_0 + \sum_{n=1}^{\infty} \left[\left(\frac{A_n}{2} + \frac{B_n}{2j} \right) e^{jn\omega_0 t} + \left(\frac{A_n}{2} - \frac{B_n}{2j} \right) e^{-jn\omega_0 t} \right] \\ &= A_0 + \sum_{n=1}^{\infty} \left(\frac{A_n - jB_n}{2} \right) e^{jn\omega_0 t} + \sum_{n=1}^{\infty} \left(\frac{A_n + jB_n}{2} \right) e^{-jn\omega_0 t} \end{aligned} \tag{190}$$

For the second summation term, if the sign of the complex sinusoid is negated and the summation limits are reversed, then we can rewrite as:

$$\begin{aligned} x(t) &= A_0 + \sum_{n=1}^{\infty} \left(\frac{A_n - jB_n}{2} \right) e^{jn\omega_0 t} + \sum_{n=-\infty}^{-1} \left(\frac{A_n + jB_n}{2} \right) e^{jn\omega_0 t} \\ &= \sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t} \end{aligned} \tag{191}$$

Writing C_n in terms of the Fourier series coefficients of Eqs. 183 and 184 gives:

$$\begin{aligned} C_0 &= A_0 \\ C_n &= (A_n - jB_n)/2 \quad \text{for} \quad n > 0 \\ C_n &= (A_n + jB_n)/2 \quad \text{for} \quad n < 0 \end{aligned} \tag{192}$$

From Eq 192, note that for $n \geq 0$:

$$C_n = \frac{A_n - jB_n}{2} = \frac{1}{T}\int_0^T x(t)\cos(n\omega_0 t)dt - j\frac{1}{T}\int_0^T x(t)\sin(n\omega_0 t)dt$$

$$= \frac{1}{T}\int_0^T x(t)[\cos(n\omega_0 t) - j\sin(n\omega_0 t)]dt \qquad (193)$$

$$= \frac{1}{T}\int_0^T x(t)e^{-jn\omega_o t}dt$$

For $n < 0$ it is clear from Eq. 192 that $C_n = C_{-n}^*$ where "*" denotes complex conjugate. Therefore we have now defined the Fourier series of a real valued signal using a complex analysis and a synthesis equation:

$$x(t) = \sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t} \qquad \text{Synthesis}$$

$$C_n = \frac{1}{T}\int_0^T x(t)e^{-jn\omega_o t}dt \qquad \text{Analysis} \qquad (194)$$

Complex Fourier Series Equations

The complex Fourier series also introduces the concept of "negative frequecies" whereby we view signals of the form $e^{j2\pi f_0}$ as a positive complex sinusoid of frequency f_0 Hz, and signals of the form $e^{-j2\pi f_0}$ as a complex sinusoid of frequency $-f_0$ Hz.

Note that the complex Fourier series is more notationally compact, and probably simpler to work with than the general Fourier series. (The "probably" depends on how clear you are in dealing with complex exponentials!) Also if the signal being analysed is in fact complex the general Fourier series of Eq. 176 (see *Fourier Series*) is insufficient but Eqs. 194 can be used. (For complex signals the coefficient relationship in Eq. 192 will not in general hold.)

Assuming the waveform being analysed is real (usually the case), then it is easy to convert C_n coefficients into A_n and B_n. Also note from Eq. 188 (see item *Fourier Series*) and Eq. 192 that:

$$M_n = \sqrt{A_n^2 + B_n^2} = 2|C_n| \qquad (195)$$

noting that $|C_n| = \sqrt{A_n^2 + B_n^2}/2$. Clearly we can also note that for the complex number C_n:

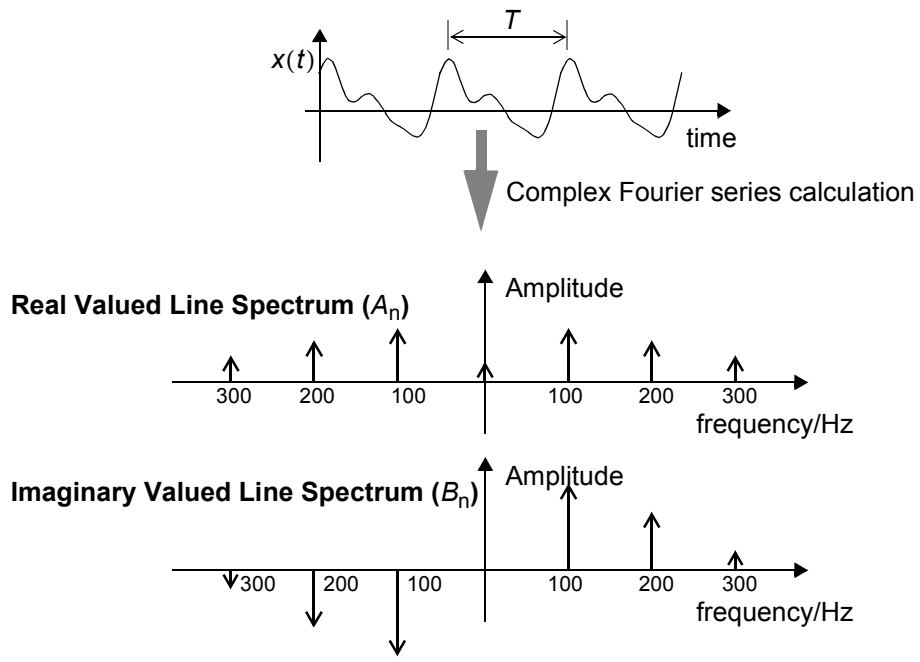
$$\angle C_n = \tan^{-1}\frac{B}{A} = \theta_n \qquad \text{i.e. } C_n = |C_n|e^{j\theta_n} \qquad (196)$$

Therefore although a complex exponential does not as such exist as a real world (single wire voltage) signal, we can easily convert from a complex exponential to a real world sinusoid simply by taking the real or imaginary part of the complex Fourier coefficients and use in the Fourier series equation (see Eq. 176, *Fourier Series*):

$$x(t) = \sum_{n=0}^{\infty} [A_n \cos(n\omega_0 t) + B_n \sin(n\omega_0 t)] \quad (197)$$

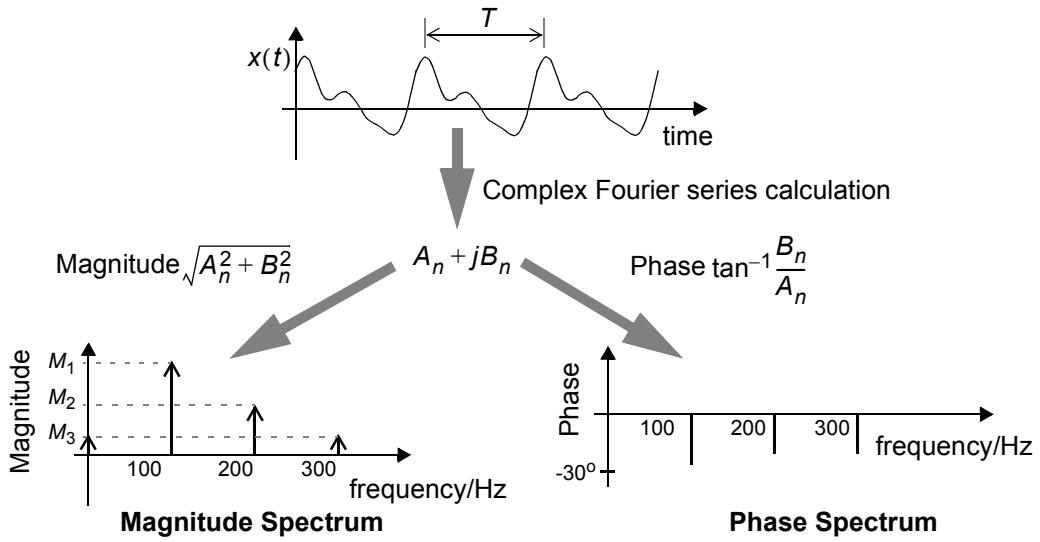
There are of course certain time domain signals which can be considered as being complex, i.e. having a separate real and imaginary components. This type of signal can be found in some digital communication systems or may be created within a DSP system to allow certain types of computation to be performed.

If a signal is decomposed into its complex Fourier series, the resulting values for the various components can be plotted as a line spectrum. As we now have both complex and real values and positive and negative frequencies, this will require two plots, one for the imaginary components and one for the real components:



The complex Fourier series line spectra. Note that there are both positive and negative frequencies, and for the complex Fourier series of a real valued signal the real line spectrum is symmetrical about $f = 0$ and the imaginary spectrum has point symmetry about the origin.

Rather than showing the real and imaginary line spectra, it is more usual to plot the magnitude spectrum and phase spectrum:



Calculating the magnitude and phase spectra from the complex Fourier series. For a real valued signal, the result will be identical, except for a magnitude scaling factor of 2, to that obtained from the amplitude phase form of the Fourier series as on page 160. As both spectra are symmetrical about the y-axis the negative frequency values are not plotted.

The "ease" of working with complex exponentials over sines and cosines can be illustrated by asking the reader to simplify the following equation to a sum of sine waves:

$$\sin(\omega_1 t)\sin(\omega_2 t) \tag{198}$$

This requires the recollection (or re-derivation!) of trigonometric identities to yield:

$$\sin(\omega_1 t)\sin(\omega_2 t) = \frac{1}{2}\cos(\omega_1 - \omega_2)t + \frac{1}{2}\cos(\omega_1 + \omega_2)t \tag{199}$$

While not particularly arduous, it is somewhat easier to simplify the following expression to a sum of complex exponentials:

$$e^{j\omega_1 t}e^{j\omega_2 t} = e^{j(\omega_1 + \omega_2)t} \tag{200}$$

Although a seemingly simple comment, this is the basis of using complex exponentials rather than sines and cosines; they make the maths easier. Of course in situations where the signal being analysed is complex, then the complex exponential Fourier series must be used.

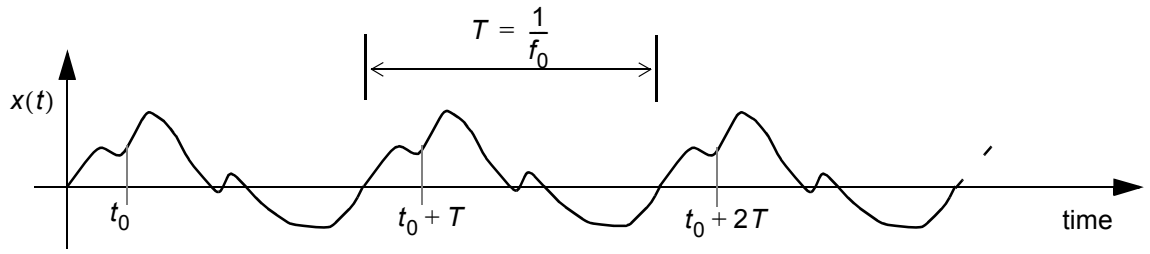
See also Discrete Fourier Transform, Fast Fourier Transform, Fast Fourier Transform - Decimation-in-Time, Fourier, Fourier Analysis, Fourier Series, Fourier Series - Amplitude/Phase Representation, Fourier Transform, Frequency Response, Impulse Response, Gibbs Phenomenon, Parseval's Theorem.

Fourier Transform: The Fourier series (rather than transform) allows a periodic signal to be broken down into a sum of real valued sine and cosine waves (in the case of a real valued signal) or more generally a sum of complex exponentials. However most signals are aperiodic, i.e. not

periodic. Therefore the Fourier transform was derived in order to analyse the frequency content of an aperiodic signal.

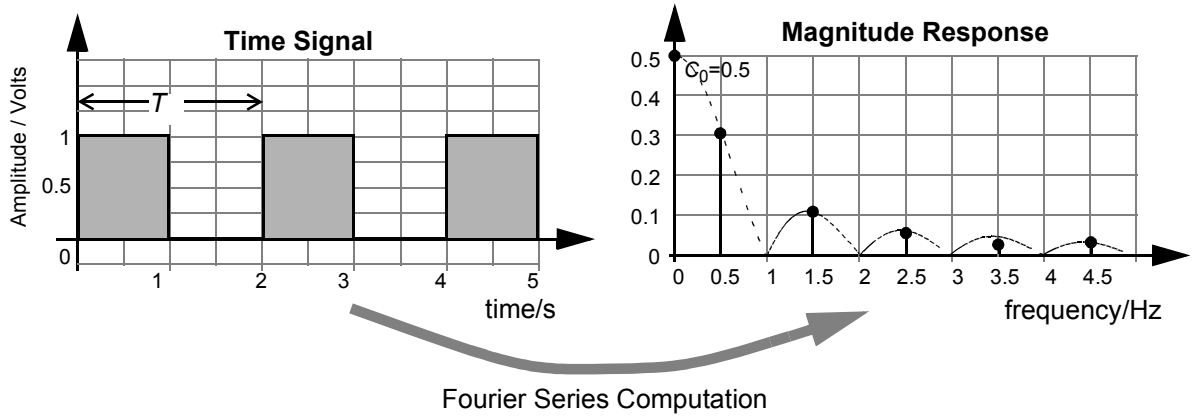
Consider the complex Fourier series of a periodic signal:

$$x(t) = \sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t}$$
$$C_n = \frac{1}{T}\int_0^T x(t) e^{-jn\omega_o t} dt \qquad (201)$$



A periodic signal $x(t)$ with period T. The fundamental frequency, f_0 is calculated simply as $f_0 = 1/T$. Clearly $x(t_0) = x(t_0 + T) = x(t_0 + 2T)$.

The period of the signal has been identified as T and the fundamental frequency is $f_0 = 1/T$. Therefore the Fourier series harmonics occur at frequencies $f_0, 2f_0, 3f_0, \ldots$.



Magnitude response of a (periodic) square wave. The phase response is zero for all components. The fundamental period is $T = 2$ and therefore the fundamental frequency is $f_0 = 1/2 = 0.5$ Hz and harmonics are therefore 0.5 Hz apart when the Fourier series is calculated.

For the above square wave we can calculate the Fourier series using Eq. 201 as:

$$C_0 = \frac{1}{T}\int_0^T s(t)dt = \frac{1}{2}\int_0^1 1\,dt = \frac{t}{2}\Big|_0^1 = \frac{1}{2} \tag{202}$$

$$\begin{aligned} C_n &= \frac{1}{T}\int_0^T s(t)e^{-j\omega_0 nt}dt = \frac{1}{2}\int_0^1 e^{-j\pi nt}dt = \frac{e^{-j\pi nt}}{-2j\pi n}\Big|_0^1 = \frac{e^{-j\pi n}-1}{-2j\pi n} \\ &= \left(\frac{e^{\frac{j\pi n}{2}} - e^{\frac{-j\pi n}{2}}}{2j\pi n}\right)e^{\frac{-j\pi n}{2}} = \frac{\sin \pi n/2}{\pi n}e^{\frac{-j\pi n}{2}} \end{aligned} \tag{203}$$

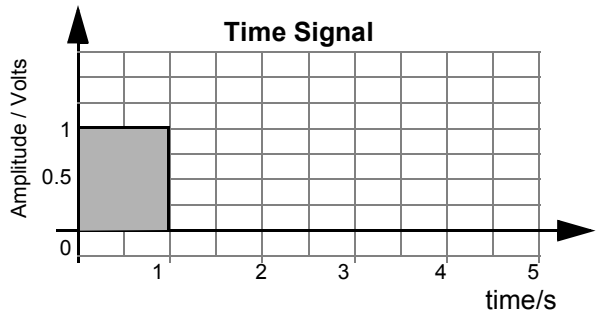
recalling that $\sin x = (e^{jx} - e^{-jx})/2j$

Noting that $e^{-j\pi n/2} = \cos \pi n/2 - j\sin \pi n/2 = 0, j$ or $-j$ (depending on the value of n) and recalling from Eq. 190 and 191 (see *Fourier Series*) that $C_n = A_n + jB_n$ then the square wave can be decomposed into a sum of harmonically related sine waves of amplitudes:

$$\begin{aligned} A_0 &= 1/2 \\ A_n &= \begin{cases} 1/n\pi \text{ for odd } n \\ 0 \text{ for even } n \end{cases} \end{aligned} \tag{204}$$

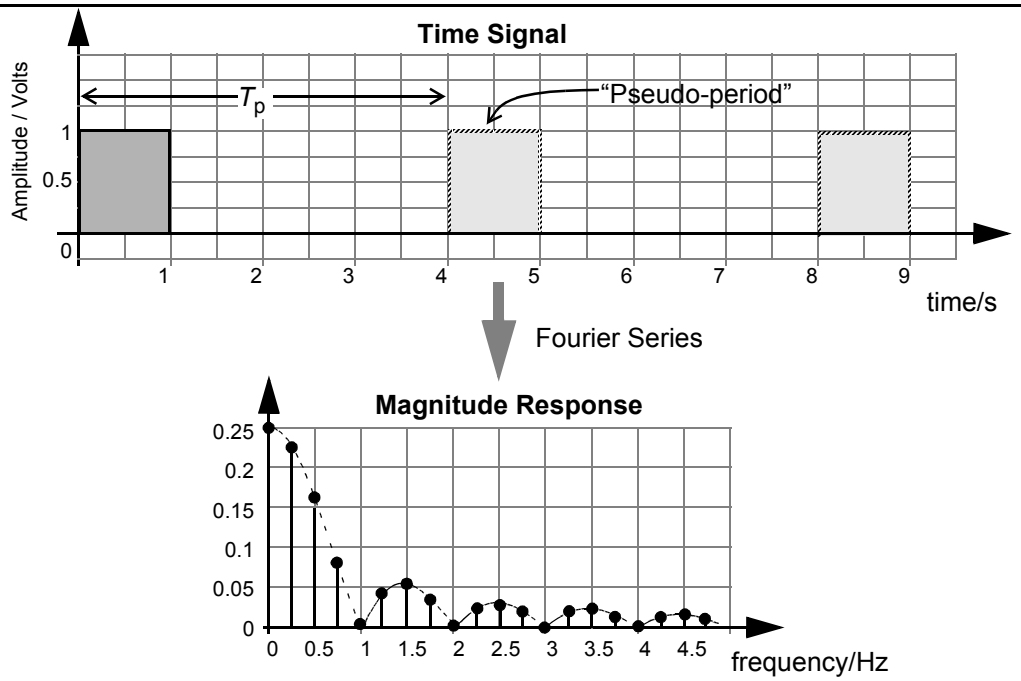
The amplitude response of the Fourier series is plotted above.

Now consider the case where the signal is aperiodic, and is in fact just a single pulse:



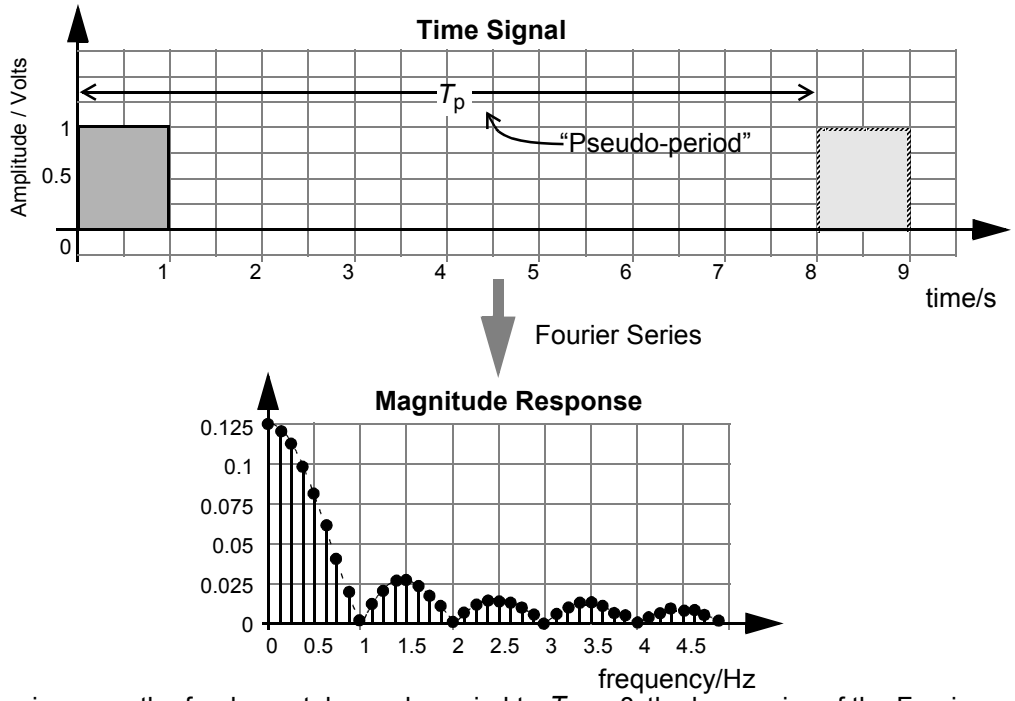
A single aperiodic pulse. This signal is most defintely not periodic and therefore the Fourier series cannot be calculated.

One way to obtain “some” information on the sinusoidal components comprising this aperiodic signal would be to assume the existence of a periodic “relative” or “pseudo-period” of this signal:



A periodic signal that is clearly a relative of the single pulse aperiodic signal. By adding the pseudo-periods we essentially assume that the single pulse of interest is a periodic signal and therefore we can now use the Fourier series tools to analyse. The fundamental period, $T_p = 4$ and therefore the harmonics of the Fourier series are placed $f_0 = 0.25$ Hz apart.

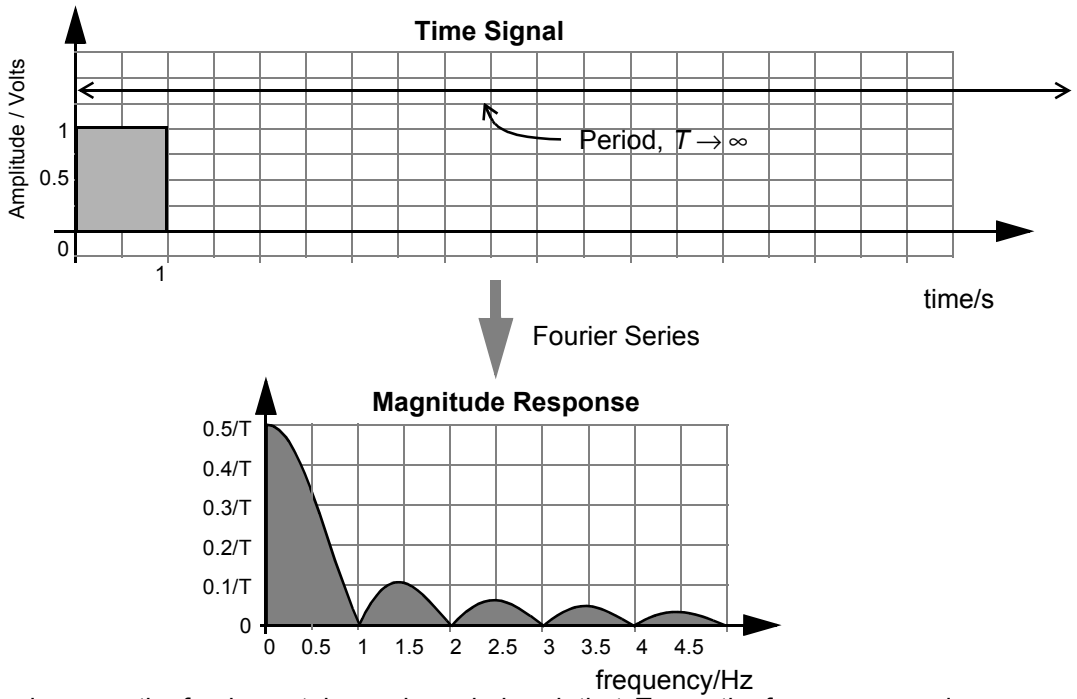
If we assumed that the “periodicity” of the pulse was even longer, say 8 seconds, then the spacing between the signal harmonics would further decrease:



If we increase the fundamental pseudo-period to $T_p = 8$ the harmonics of the Fourier series are more closely spaced at $f_0 = 1/8 = 0.125$ Hz apart. The magnitude of all the harmonics proportionally decreases with the increase in the pseudo-period. This is expected since the power of the signal decreases as the number of harmonics decreases.

If we further assumed that the period of the signal was such that $T \to \infty$ then $f_0 \to \infty$ and given the finite energy in the signal, the magnitude of each of the Fourier series sine waves will tend to zero given that the harmonics are now so closely spaced! Hence if we multiply the magnitude response

by T and plot the Fourier series we have now realised a graphical interpretation of the Fourier transform:



If we increase the fundamental pseudo-period such that $T \to \infty$ the frequency spacing between the harmonics of the Fourier series tends to zero, i.e. $f_0 \to 0$. Note that the magnitude of the Fourier series components are scaled proportionally down by the value of the "pseudo" period and in the limit as $T \to \infty$ will tend to zero. Hence the y-axis is plotted as $1/T$.

To realise the mathematical version of the Fourier transform first define a new function based on the general Fourier series of Eq. 201 such that:

$$X(f) = \frac{C_n}{f_0} = C_n T \tag{205}$$

then:

$$\begin{aligned} x(t) &= \sum_{n=-\infty}^{\infty} C_n e^{j2\pi n f_0 t} \\ X(f) &= \int_{-T/2}^{T/2} x(t) e^{-j2\pi n f_0 t}\, dt = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \end{aligned} \tag{206}$$

where nf_0 becomes the continuous variable f as $f_0 \to 0$ and $n \to \infty$. This equation is refered to as the Fourier transform and can of course be written in terms of the angular frequency:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{207}$$

Knowing the Fourier transform of a signal, of course allows us to transform back to the original aperiodic signal:

$$\begin{aligned} x(t) &= \sum_{n=-\infty}^{\infty} C_n e^{j2\pi n f_0 t} = \sum_{n=-\infty}^{\infty} X(f) f_0 e^{j2\pi n f_0 t} = \left(\sum_{n=-\infty}^{\infty} X(f) e^{j2\pi n f_0 t} \right) f_0 \\ \Rightarrow x(t) &= \int_{-\infty}^{\infty} X(f) e^{j2\pi f t}\, df \end{aligned} \tag{208}$$

This equation is refered to as the inverse Fourier transform and can also be written in terms of the angular frequency:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t}\, d\omega \tag{209}$$

Hence we have realised the Fourier transform analysis and synthesis pair of equations:

$$\begin{aligned} x(t) &= \int_{-\infty}^{\infty} X(f) e^{j2\pi f t}\, df \quad \text{Synthesis} \\ X(f) &= \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt \quad \text{Analysis} \end{aligned}$$

Fourier Transform Pair

(210)

Therefore the Fourier transform of a continuous time signal, $x(t)$, will be a continuous function in frequency.

See also Discrete Cosine Transform, Discrete Fourier Transform, Fast Fourier Transform, Fourier Analysis, Fourier Series, Fourier Series - Complex Exponential Representation, Fourier Transform.

Forward Substitution: See *Matrix Algorithms - Forward Substitution*.

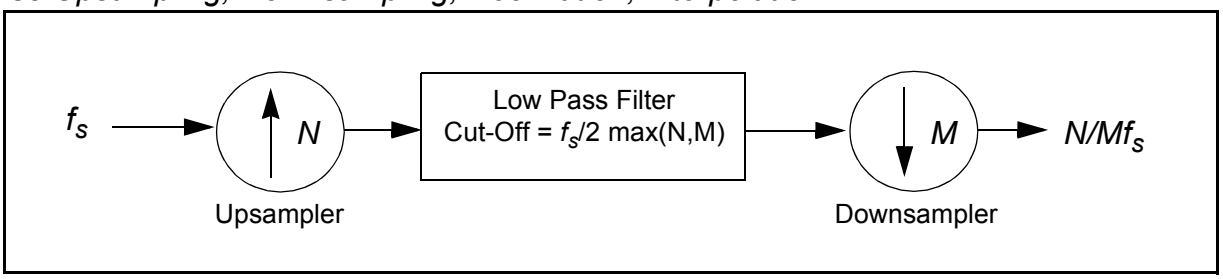
Fractals: *Fractals* can be used to define seemingly irregular 1-D signals or 2-D surfaces using, amongst other things, properties of self similarity. Self similarity occurs when the same pattern repeats itself at different scalings, and is often seen in nature. A good introduction and overview of *fractals* can be found in [86].

Fractional Binary: See *Binary Point*.

Fractional Bandwidth: A definition of (relative) bandwidth for a signal obtained by dividing the difference of the highest and lowest frequencies of the signal by its center frequency. The result is a number between 0 and 2. When this number is multiplied by 100, the relative bandwidth can be stated in terms of percentage. See also *Bandwidth.*

Fractional Delay Implementation: See *All-pass Filter - Fractional Sample Delay Implementation*.

Fractional Sampling Rate Conversion: Sometimes sampling rate conversions are needed between sampling rates that are not integer multiples of each other and therefore simple integer downsampling or upsampling cannot be performed. One method of changing sampling rate is to convert a signal back to its analog form using a DAC, then resample the signal using an ADC sampling at the required frequency. In general this is not acceptable solution as two levels of noise are introduced by the DAC and ADC Interpolation by a factor of *N*, followed by decimation by a factor of *M* results in a sampling rate change of *N/M*. The higher the values of *N* and *M*, the more computation that is required. For example to convert from CD sampling rates of 44100Hz to DAT sampling rate of 48000Hz requires upsampling by a factor of 160, and downsampling by a factor of 147. When performing *fractional sampling rate conversion* the low pass anti-alias filter associated with decimation, and the low pass filter used in interpolation can be combined into one digital filter. See also *Upsampling, Downsampling, Decimation, Interpolation*.



Frequency: *Frequency* is measured in Hertz (Hz) and gives a measure of the number of cycles per second of a signal. For example if a sine wave has a *frequency* of 300Hz, this means that the signal has 300 single wavelength cycles in one second. Square waves also can be assigned a frequency that is defined as *1/T* where *T* is the period of one cycle of the square wave. See also *Sine Wave.*

Frequency Domain Adaptive Filtering: The LMS (and other adaptive algorithms) can be configured to operate of time series data that has been transformed into the frequency domain [53], [131].

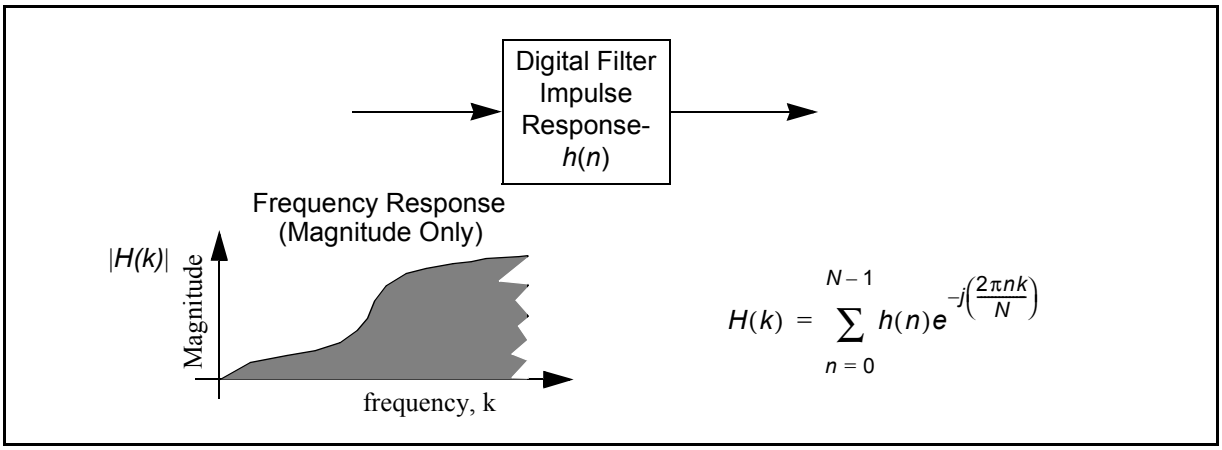
Frequency, Logarithmic: See *Logarithmic Frequency.*

Frequency Modulation: One of the three ways of modulating a sine wave signal to carry information. The sine wave or carrier has its frequency changed in accordance with the information signal to be transmitted. See also *Amplitude Modulation, Phase Modulation*.

Frequency Range of Hearing: The *frequency range of hearing* typically goes from around 20Hz to up to 20kHz in healthy young people. For adults the upper range of hearing is more likely to be in the range 11-16kHz as age erodes the high frequency sensitivity. The threshold of hearing varies over the frequency range, with the most sensitive portion being from around 1-5kHz, where speech frequencies occur. Low frequencies, below 20Hz, are tactile and only audible at very high sound pressure levels. Also listening to frequencies below 20Hz does not produce any further perception of reducing pitch. Inaudible sound below the lowest perceptible frequency is termed infrasound, and above the highest perceptible frequency, is known as ultrasound.

Discrimination between tones at similar frequencies (the JND - just noticeable difference or DL - Difference Limen), depends on a number of factors such as the frequency, sound pressure level (SPL), and sound duration. The ear can discriminate by about 1Hz for frequencies in the range 1-2kHz where the SPL is about 20dB above the threshold of hearing, and the duration is at least 1/4 seconds [30]. See also *Audiogram, Audiometry, Auditory Filters, Beat Frequencies, Binaural Beats, Difference Limen, Ear, Equal Loudness Contours, Hearing Aids, Hearing Impairment, Hearing Level, Infrasound, Sensation Level, Sound Pressure Level, Spectral Masking, Temporal Masking, Threshold of Hearing, Ultrasound*.

Frequency Response: The *frequency response* a system defines how the magnitude and phase of signal components at different frequencies will be changed as the signal passes through, or is convolved with a linear system. For example the frequency response of a digital filter may attenuate low frequency magnitudes, but amplify those at high frequencies. The frequency response of a linear system is calculated by taking the discrete Fourier transform (DFT) of the impulse response or evaluating the z-transform of the linear system for $z = e^{j\omega} = e^{j2\pi f}$. See also Discrete Fourier Transform, Fast Fourier Transform. .



Frequency Shift Keying (FSK): A digital modulation technique in which the information bits are encoded in the frequency of a symbol. Typically, the frequencies are chosen so that the symbols are orthogonal over the symbol period. *FSK* demodulation can be either coherent (phase of carrier signal known) or noncoherent (phase of carrier signal unknown). Given a symbol period of *T* seconds, signals separated in frequency by *1/T* Hz will be orthogonal and will have continuous phase. Signals separated by *1/(2T)* Hz will be orthogonal (if demodulated coherently) but will result in phase discontinuities. See also *Amplitude Shift Keying, Continuous Phase Modulation, Minimum Shift Keying, Phase Shift Keying.*

Frequency Transformation: The transformation of any time domain signal into the frequency domain.

Frequency Weighting Curves: See Sound Pressure Level Weighting Curves.

Frobenius Norm: See *Matrix Properties - Norm.*

Formants: The vocal tract (comprising throat, mouth and lips) can act as an acoustics resonator with more than one resonant frequency. These resonant frequencies are known as *formants* and they change in frequency while we move tongue and lips in the process of joining speech sounds together (articulation).

Four Wire Circuit: A circuit containing two pairs of wires (or their logical equivalent) for simultaneous (full duplex) two transmission. See also *Two Wire Channel, Full Duplex, Half Duplex, Simplex*.

Fricatives: One of the elementary sounds of speech, namely plosives, *fricatives*, sibilant fricative, semi-vowels, and nasals. *Fricatives* are formed from the lower lip and teeth with air through as when "f" is used in the word "fin". See also *Nasals, Plosives, Semi-vowels, and Sibilant Fricatives*.

Full Adder: The *full adder* is the basic single bit arithmetic building block for design of multibit binary adders, multipliers and arithmetic logic units. The *full adder* has three single bit inputs and two single bit outputs:

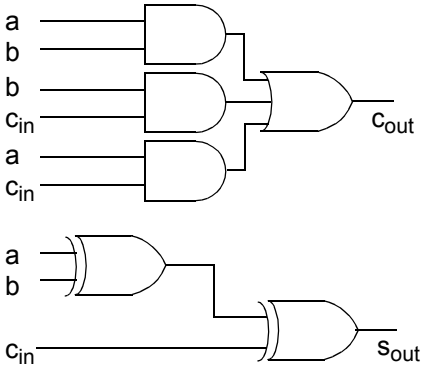
a	*b*	c_{in}	c_{out}	s_{out}
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Truth Table

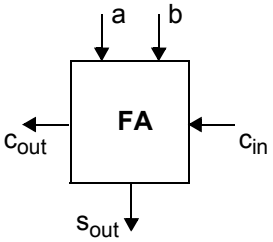
$$c_{out} = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc = ab + bc + ac$$

$$s_{out} = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc = (a \oplus b) \oplus c$$

Boolean Algebra



Logic Circuit



Symbol

Boolean Algebra: *(a+b)* represents *(a OR b); (ab)* represents (*a AND b); $a \oplus b$* represents *(a Exclusive-OR b).* The full adder (FA) simply adds three bits (0 or 1) together to produce a sum bit, s_{out} and carry bit, c_{out}

See also *Arithmetic Logic Unit, Parallel Adder, Parallel Multiplier, DSP Processor*.

Full Duplex: Pertaining to the capability to send and receive simultaneously. See also *Half Duplex, Simplex*.

Fundamental Frequency: The name of the lowest (and usually) dominant frequency component which has associated with it various harmonics (integer multiples of the frequency). In music for example the fundamental frequency identifies the note being played, and the various harmonics (and occasionally sub-harmonics) give the note its rich characteristic quality pertaining to the instrument being played. See also Fourier Series, *Harmonics, Music, Sub-Harmonic, Western Music Scale*.

Fundamental Period: See also Fourier Series.

Fuzzy Logic: A mathematical set theory which allows systems to be described in natural language rules. Binary for example uses only two level logic: 0 and 1. *Fuzzy logic* would still have the levels 0 and 1, but it would also be capable of describing all logic levels in between perhaps ranging through: almost definitely low, probably low, maybe high or low, probably high, to almost definitely high. Control of systems defined by *fuzzy logic* are currently being implemented in conjunction with

DSP algorithms. Essentially *fuzzy logic* is a technique for representing information and combining objective knowledge (such as mathematical models and precise definitions) with subjective knowledge (a linguistic description of a problem). One advantage often cited about fuzzy systems is that they can produce results almost as good as an “optimum” system, but they are much simpler to implement. A good introduction, with tutorial papers, can be found in [63].

G

G-Series Recommendations: The *G-series recommendations* from the International Telecommunication (ITU), advisory committee on telecommunications (denoted ITU-T, and formerly known as CCITT) propose a number of standards for transmission systems and media, digital systems and networks. From a DSP perspective the G164/5/6/7 define aspects of echo and acoustic echo cancellation, and some of the G.7XX define various coding and compression schemes which underpin digital audio telecommunication. The ITU-T *G-series recommendations* (http://www.itu.ch) can be summarised as:

- G.100 Definitions used in Recommendations on general characteristics of international telephone connections and circuits.
- G.101 The transmission plan.
- G.102 Transmission performance objectives and Recommendations.
- G.103 Hypothetical reference connections.
- G.105 Hypothetical reference connection for crosstalk studies.
- G.111 Loudness ratings (LRs) in an international connection.
- G.113 Transmission impairments.
- G.114 One-way transmission time.
- G.117 Transmission aspects of unbalance about earth (definitions and methods).
- G.120 Transmission characteristics of national networks.
- G.121 Loudness ratings (LRs) of national systems.
- G.122 Influence of national systems on stability and talker echo in international connections.
- G.123 Circuit noise in national networks.
- G.125 Characteristics of national circuits on carrier systems.
- G.126 Listener echo in telephone networks.
- G.132 Attenuation distortion.
- G.133 Group-delay distortion.
- G.134 Linear crosstalk.
- G.135 Error on the reconstituted frequency.
- G.141 Attenuation distortion.
- G.142 Transmission characteristics of exchanges.
- G.143 Circuit noise and the use of Companders.
- G.151 General performance objectives applicable to all modern international circuits and national extension circuits.
- G.152 Characteristics appropriate to long-distance circuits of a length not exceeding 2500 km.
- G.153 Characteristics appropriate to international circuits more than 2500 km in length.
- G.162 Characteristics of Companders for telephony.
- G.164 Echo suppressors.
- G.165 Echo cancellers.
- G.166 Characteristics of syllabic Companders for telephony on high capacity long distance systems.
- G.167 Acoustic echo controllers.
- G.172 Transmission plan aspects of international conference calls.
- G.173 Transmission planning aspects of the speech service in digital public land mobile networks.
- G.174 Transmission performance objectives for terrestrial digital wireless systems using portable terminals to access the PSTN.
- G.180 Characteristics of N + M type direct transmission restoration systems for use on digital and analogue sections, links or equipment.
- G.181 Characteristics of 1 + 1 type restoration systems for use on digital transmission links.
- G.191 Software tools for speech and audio coding standardization.
- G.211 Make-up of a carrier link.
- G.212 Hypothetical reference circuits for analogue systems.
- G.213 Interconnection of systems in a main repeater station.
- G.214 Line stability of cable systems.

G.215 Hypothetical reference circuit of 5000 km for analogue systems.
G.221 Overall recommendations relating to carrier-transmission systems.
G.222 Noise objectives for design of carrier-transmission systems of 2500 km.
G.223 Assumptions for the calculation of noise on hypothetical reference circuits for telephony.
G.224 Maximum permissible value for the absolute power level (power referred to one milliwatt) of a signalling pulse.
G.225 Recommendations relating to the accuracy of carrier frequencies.
G.226 Noise on a real link.
G.227 Conventional telephone signal.
G.228 Measurement of circuit noise in cable systems using a uniform-spectrum random noise loading.
G.229 Unwanted modulation and phase jitter.
G.230 Measuring methods for noise produced by modulating equipment and through-connection filters.
G.231 Arrangement of carrier equipment.
G.232 12-channel terminal equipments.
G.233 Recommendations concerning translating equipments.
G.241 Pilots on groups, supergroups, etc.
G.242 Through-connection of groups, supergroups, etc.
G.243 Protection of pilots and additional measuring frequencies at points where there is a through-connection.
G.322 General characteristics recommended for systems on symmetric pair cables.
G.325 General characteristics recommended for systems providing 12 telephone carrier circuits on a symmetric cable pair [(12+12) systems].
G.332 12 MHz systems on standardized 2.6/9.5 mm coaxial cable pairs.
G.333 60 MHz systems on standardized 2.6/9.5 mm coaxial cable pairs.
G.334 18 MHz systems on standardized 2.6/9.5 mm coaxial cable pairs.
G.341 1.3 MHz systems on standardized 1.2/4.4 mm coaxial cable pairs.
G.343 4 MHz systems on standardized 1.2/4.4 mm coaxial cable pairs.
G.344 6 MHz systems on standardized 1.2/4.4 mm coaxial cable pairs.
G.345 12 MHz systems on standardized 1.2/4.4 mm coaxial cable pairs.
G.346 18 MHz systems on standardized 1.2/4.4 mm coaxial cable pairs.
G.352 Interconnection of coaxial carrier systems of different designs.
G.411 Use of radio-relay systems for international telephone circuits.
G.421 Methods of interconnection.
G.422 Interconnection at audio-frequencies.
G.423 Interconnection at the baseband frequencies of frequency-division multiplex radio-relay systems.
G.431 Hypothetical reference circuits for frequency-division multiplex radio-relay systems.
G.441 Permissible circuit noise on frequency-division multiplex radio-relay systems.
G.442 Radio-relay system design objectives for noise at the far end of a hypothetical reference circuit with reference to telegraphy transmission.
G.451 Use of radio links in international telephone circuits.
G.473 Interconnection of a maritime mobile satellite system with the international automatic switched telephone service; transmission aspects.
G.601 Terminology for cables.
G.602 Reliability and availability of analogue cable transmission systems and associated equipments (10)
G.611 Characteristics of symmetric cable pairs for analogue transmission.
G.612 Characteristics of symmetric cable pairs designed for the transmission of systems with bit rates of the order of 6 to 34 Mbit/s.
G.613 Characteristics of symmetric cable pairs usable wholly for the transmission of digital systems with a bit rate of up to 2 Mbits.
G.614 Characteristics of symmetric pair star-quad cables designed earlier for analogue transmission systems and being used now for digital system transmission at bit rates of 6 to 34 Mbit/s.
G.621 Characteristics of 0.7/2.9 mm coaxial cable pairs.
G.622 Characteristics of 1.2/4.4 mm coaxial cable pairs.
G.623 Characteristics of 2.6/9.5 mm coaxial cable pairs.
G.631 Types of submarine cable to be used for systems with line frequencies of less than about 45 MHz.
G.650 Definition and test methods for the relevant parameters of single-mode fibres.
G.651 Characteristics of a 50/125 μm multimode grades index optical fibre cable.

G.652 Characteristics of a single-mode optical fibre cable.
G.653 Characteristics of a dispersion-shifted single-mode optical fibre cable.
G.654 Characteristics of a 1550 nm wavelength loss-minimized single-mode optical fibre cable.
G.661 Definition and test methods for relevant generic parameters of optical fibre amplifiers.
G.662 Generic characteristics of optical fibre amplifier devices and sub-systems.
G.701 Vocabulary of digital transmission and multiplexing, and pulse code modulation (PCM) terms.
G.702 Digital hierarchy bit rates.
G.703 Physical/electrical characteristics of hierarchical digital interfaces.
G.704 Synchronous frame structures used at primary and secondary hierarchical levels.
G.705 Characteristics required to terminate digital links on a digital exchange.
G.706 Frame alignment and cyclic redundancy check (CGC) procedures relating to basic frame structures defined in Recommendation G.704.
G.707 Synchronous digital hierarchy bit rates.
G.708 Network node interface for the synchronous digital hierarchy.
G.709 Synchronous multiplexing structure.
G.711 Pulse code modulation (PCM) of voice frequencies.
G.712 Transmission performance characteristics of pulse code modulation.
G.720 Characterization of low-rate digital voice coder performance with non-voice signals.
G.722 7 kHz audio-coding within 64 kbit/s; Annex A: Testing signal-to-total distortion ratio for kHz audio-codecs at 64 kbit/s.
G.724 Characteristics of a 48-channel low bit rate encoding primary multiplex operating at 1544 kbit/s.
G.725 System aspects for the use of the 7 kHz audio codec within 64 kbit/s.
G.726 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM). Annex A: Extensions of Recommendation G.726 for use with uniform-quantized input and output.
G.727 5-, 4-, 3- and 2-bits sample embedded adaptive differential pulse code modulation (ADPCM).
G.728 Coding of speech at 16 kbit/s using low-delay code excited linear prediction. Annex G to Coding of speech at 16 kbit/s using low-delay code excited linear prediction: 16 kbit/s fixed point specification.
G.731 Primary PCM multiplex equipment for voice frequencies.
G.732 Characteristics of primary PCM multiplex equipment operating at 2048 kbit/s.
G.733 Characteristics of primary PCM multiplex equipment operating at 1544 kbit/s.
G.734 Characteristics of synchronous digital multiplex equipment operating at 1544 kbit/s.
G.735 Characteristics of primary PCM multiplex equipment operating at 2048 kbit/s and offering synchronous digital access at 384 kbit/s and/or 64 kbit/s.
G.736 Characteristics of a synchronous digital multiplex equipment operating at 2048 kbit/s.
G.737 Characteristics of an external access equipment operating at 2048 kbit/s offering synchronous digital access at 384 kbit/s and/or 64 kbit/s.
G.738 Characteristics of primary PCM multiplex equipment operating at 2048 kbit/s and offering synchronous digital access at 320 kbit/s and/or 64 kbit/s.
G.739 Characteristics of an external access equipment operating at 2048 kbit/s offering synchronous digital access at 320 kbit/s and/or 64 kbit/s.
G.741 General considerations on second order multiplex equipments.
G.742 Second order digital multiplex equipment operating at 8448 kbit/s and using positive justification.
G.743 Second order digital multiplex equipment operating at 6312 kbit/s and using positive justification.
G.744 Second order PCM multiplex equipment operating at 8448 kbit/s.
G.745 Second order digital multiplex equipment operating at 8448 kbit/s and using positive/zero/negative justification.
G.746 Characteristics of second order PCM multiplex equipment operating at 6312 kbit/s.
G.747 Second order digital multiplex equipment operating at 6312 kbit/s and multiplexing three tributaries at 2048 kbit/s.
G.751 Digital multiplex equipments operating at the third order bit rate of 34368 kbit/s and the fourth order bit rate of 139264 kbit/s and using positive justification.
G.752 Characteristics of digital multiplex equipments based on a second order bit rate of 6312 kbit/s and using positive justification.
G.753 Third order digital multiplex equipment operating at 34368 kbit/s and using positive/zero/negative justification.
G.754 Fourth order digital multiplex equipment operating at 139264 kbit/s and using positive/zero/negative justification.

G.755 Digital multiplex equipment operating at 139264 kbit/s and multiplexing three tributaries at 44736 kbit/s.
G.761 General characteristics of a 60-channel transcoder equipment.
G.762 General characteristics of a 48-channel transcoder equipment.
G.763 Summary of Recommendation G.763.
G.764 Voice packetizationpacketized voice protocols.
G.765 Packet circuit multiplication equipment.
G.766 Facsimile demodulation/remodulation for DCME.
G.772 Protected monitoring points provided on digital transmission systems.
G.773 Protocol suites for Q-interfaces for management of transmission systems.
G.774 Synchronous Digital Hierarchy (SDH) management information model for the network element view. G.774.01: Synchronous digital hierarchy (SDH) performance monitoring for the network element view. G.774.02: Synchronous digital hierarchy (SDH) configuration of the payload structure for the network element view. G.774.03: Synchronous digital hierarchy (SDH) management of multiplex-section protection for the network element view.
G.775 Loss of signal (LOS) and alarm indication signal (AIS) defect detection and clearance criteria.
G.780 Vocabulary of terms for synchronous digital hierarchy (SDH) networks and equipment.
G.781 Structure of Recommendations on equipment for the synchronous digital hierarchy (SDH).
G.782 Types and general characteristics of synchronous digital hierarchy (SDH) equipment.
G.783 Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks.
G.784 Synchronous digital hierarchy (SDH) management.
G.791 General considerations on transmultiplexing equipments.
G.792 Characteristics common to all transmultiplexing equipments.
G.793 Characteristics of 60-channel transmultiplexing equipments.
G.794 Characteristics of 24-channel transmultiplexing equipments.
G.795 Characteristics of codecs for FDM assemblies.
G.796 Characteristics of a 64 kbit/s cross-connect equipment with 2048 kbit/s access ports.
G.797 Characteristics of a flexible multiplexer in a plesiochronous digital hierarchy environment.
G.801 Digital transmission models.
G.802 Interworking between networks based on different digital hierarchies and speech encoding laws.
G.803 Architectures of transport networks based on the synchronous digital hierarchy (SDH).
G.804 ATM cell mapping into plesiochronous digital hierarchy (PDH).
G.821 Error performance of an international digital connection forming part of an integrated services digital network.
G.822 Controlled slip rate objectives on an international digital connection.
G.823 The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy.
G.824 The control of jitter and wander within digital networks which are based on the 1544 kbit/s hierarchy.
G.825 The control of jitter and wander within digital networks which are based on the Synchronous Digital Hierarchy (SDH).
G.826 Error performance parameters and objectives for international, constant bit rate digital paths at or above the primary rate.
G.831 Management capabilities of transport networks based on the Synchronous Digital Hierarchy (SDH).
G.832 Transport of SDH elements on PDH networks: Frame and multiplexing structures.
G.901 General considerations on digital sections and digital line systems.
G.911 Parameters and calculation methodologies for reliability and availability of fibre optic systems.
G.921 Digital sections based on the 2048 kbit/s hierarchy.
G.931 Digital line sections at 3152 kbit/s.
G.950 General considerations on digital line systems.
G.951 Digital line systems based on the 1544 kbit/s hierarchy on symmetric pair cables.
G.952 Digital line systems based on the 2048bit/s hierarchy on symmetric pair cables.
G.953 Digital line systems based on the 1544 kbit/s hierarchy on coaxial pair cables.
G.954 Digital line systems based on the 2048 kbit/s hierarchy on coaxial pair cables.
G.955 Digital line systems based on the 1544 kbit/s and the 2048 kbit/s hierarchy on optical fibre cables.
G.957 Optical interfaces for equipments and systems relating to the synchronous digital hierarchy.
G.958 Digital line systems based on the synchronous digital hierarchy for use on optical fibre cables.
G.960 Access digital section for ISDN basic rate access.
G.961 Digital transmission system on metallic local lines for ISDN basic rate access.
G.962 Access digital section for ISDN primary rate at 2048 kbit/s.

G.963 Access digital section for ISDN primary rate at 1544 kbit/s.
G.964 V-Interfaces at the digital local exchange (LE)V5.1-Interface (based on 2048 kbit/s) for the support of access network (AN).
G.965 V-Interfaces at the digital local exchange (LE)V5.2 interface (based on 2048 kbit/s) for th support of Access Network (AN).
G.971 General features of optical fibre submarine cable systems.
G.972 Definition of terms relevant to optical fibre submarine cable systems.
G.974 Characteristics of regenerative optical fibre submarine cable systems.
G.981 PDH optical line systems for the local network.

For additional detail consult the appropriate standard document or contact the ITU. See also *International Telecommunication Union, ITU-T Recommendations, Standards*.

Gabor Spectrogram: An algorithm to transform signals from the time domain to the joint time-frequency domain (similar to the Short Time FFT spectrogram). The Gabor is most useful for analyzing signals who frequency content is time varying, but which does not show up on conventional spectrogram methods. For example in a particular jet engine the casing vibrates at 50Hz when running at full speed. If the frequency actually fluctuates about ±1Hz around 50Hz, then when using the conventional FFT the fluctuations may not have enough energy to be detected or may be smeared due to windowing effects. The Gabor spectrogram on the other hand should be able to highlight the fluctuations.

Gain: An increase in the voltage, or power level of a signal usually accomplish by an amplifier. Gain is expressed as a factor, or in dB. See also *Amplifier*.

Gauss Transform: See *Matrix Decompositions - Gauss Transform*.

Gaussian Distribution: See *Random Variable*.

Gaussian Elimination: See *Matrix Decompositions - Gaussian Elimination*.

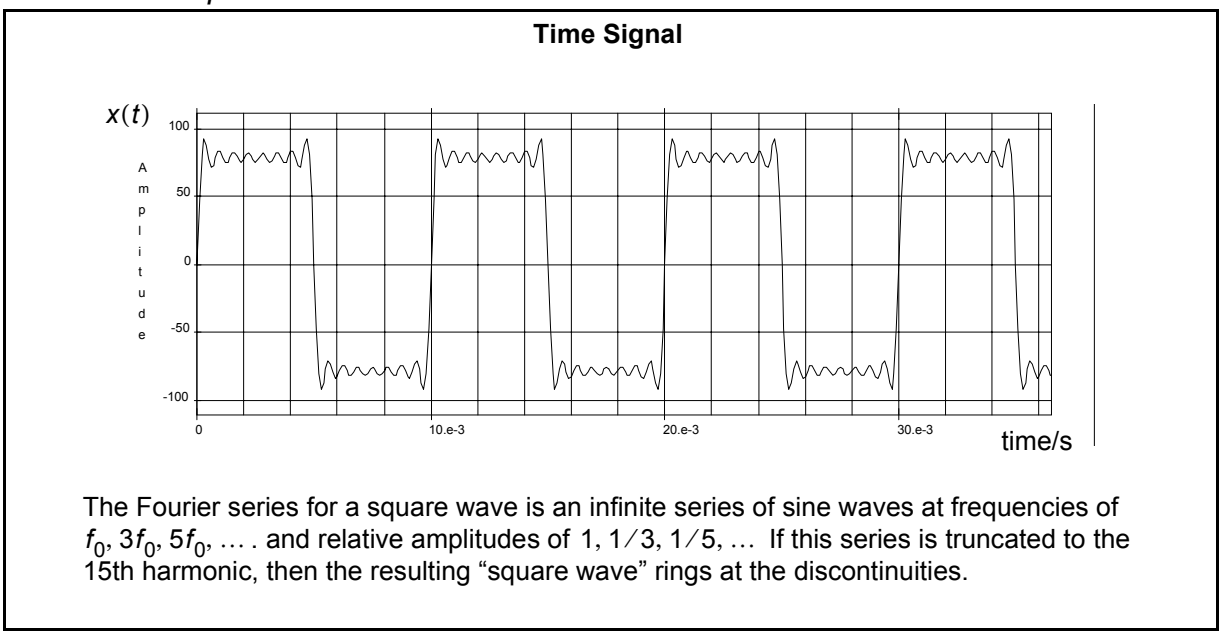
Gibbs Phenomenon: The Fourier series for a periodic signal with (almost) discontinuities will tend to an infinite series. If the signal is approximated using a finite series of harmonics then the reconstructed signal will tend to oscillate near or on the discontinuities. For example, the Fourier series of a signal, $x(t)$, is given by:

$$x(t) = \sum_{n=0}^{\infty} A_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{\infty} B_n \sin\left(\frac{2\pi nt}{T}\right) \tag{211}$$

For a signal such as a square wave, the series will be infinite. If however we try to produce the signal using just the first few Fourier series coefficients up to M:

$$x(t) = \sum_{n=0}^{M} A_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{M} B_n \sin\left(\frac{2\pi nt}{T}\right) \tag{212}$$

then “ringing” will be seen near the discontinuties since to adequately represent these parts of the waveform we require the high frequency components which have been truncated. This ringing is refered to as *Gibb’s phenonmenon*.



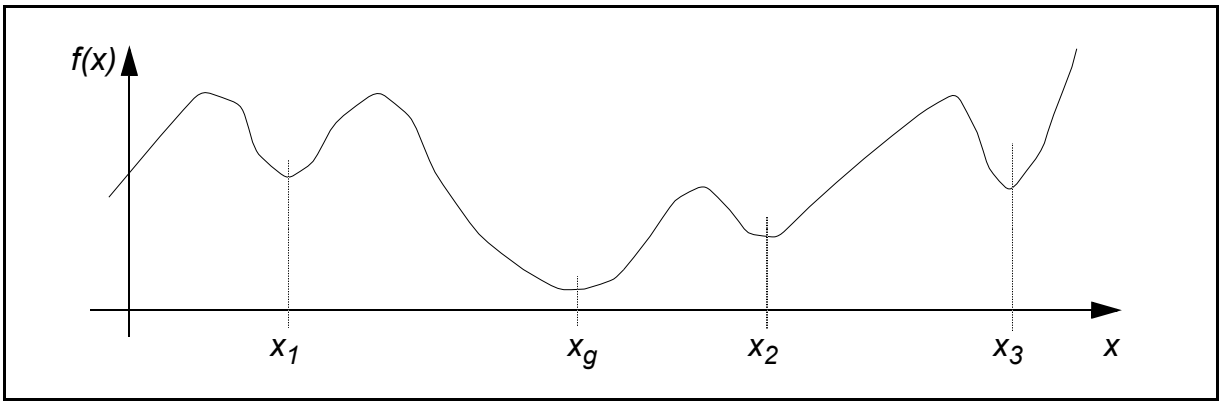
The Fourier series for a square wave is an infinite series of sine waves at frequencies of f_0, $3f_0$, $5f_0$, … . and relative amplitudes of 1, 1⁄3, 1⁄5, … If this series is truncated to the 15th harmonic, then the resulting “square wave” rings at the discontinuities.

See also Discrete Fourier Transform, Fourier Series, Fourier Series - Amplitude/Phase Representation, Fourier Series - Complex Exponential Representation, Fourier Transform.

Given’s Rotations: See *Matrix Decompositions - Given’s Rotations*.

Global Information Infrastructure (GII): The *Global Information Infrastructure* will be jointly defined by the International Organization for Standards (ISO), International Electrotechnical Committee (IEC) and the International Telecommunication Union (ITU). The ISO, IEC and ITU have all defined various standards that have direct relevance to interchange of graphics, audio, video and data information via computer and telephone networks and all therefore have a relevant role to play in the definition of the GII.

Global Minimum: The *global minimum* is the smallest value taken on by that function. For example for the function, *f(x)*, the global minimum is at $x = x_g$. The minima are x_1, x_2 and x_3 are termed local minima:



The existence of local minima can cause problems when using a gradient descent based adaptive algorithm. In these cases, the algorithm can get stuck in a local minimum. This is not a problem when the cost function is quadratic in the parameter of interest (e.g., the filter coefficients), since

quadratic functions (such as a parabola) have a unique minimum (or maximum) or, worst case, a set of continuous minima that all give the same cost. See also *Hyperparaboloid, Local Minima, Adaptive IIR Filters, Simulated Annealing*.

Glue Logic: To connect different chips on printed circuit boards (PCBs) it is often necessary to use buffers, inverters, latches, logic gates etc. These components are often referred to a *glue logic*. Many DSP chip designers pride themselves in having eliminated *glue logic* for chip interfacing, especially between D/A and A/D type chips.

Golden Ears: A term often used to describe a person with excellent hearing, both in terms of frequency range and threshold of hearing. *Golden ear* individuals can be in demand from recording studios, audio equipment manufacturers, loudspeaker manufacturers and so on. Although a necessary qualification for *golden ears* is excellent hearing, these individuals most probably learn their trade from many years of audio industry experience. It would be expected that a *golden ears* individual could "easily" distinguish Compact Disc (CD) from analog records. The big irony is that *golden ear*ed individuals cannot distinguish recordings of REO Speedwagon from those of Styx. See also *Audiometry, Compact Disc, Frequency Range of Hearing, Threshold of Hearing*.

Goertzel's Algorithm: Goertzel's algorithm is used to calculate if a frequency component is present at a particular frequency bin of a discrete Fourier transform (DFT). Consider the DFT equation calculating the discrete frequency domain representation, $X(m)$, of N samples of a discrete time signal $x(k)$:

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{-j\left(\frac{2\pi nm}{N}\right)} \quad \text{, for all } k = 0 \text{ to } N-1 \tag{213}$$

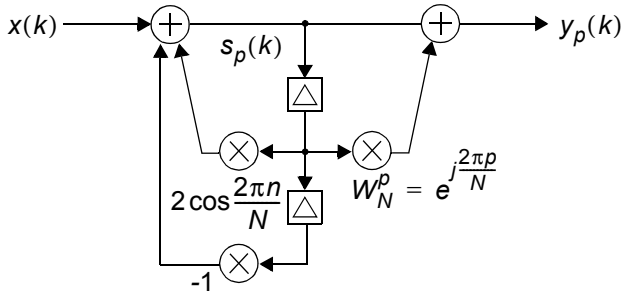
This computation requires N^2 complex multiply accumulates (CMACs), and the frequency representation will have a resolution of f_s / N Hz. If we require to calculate the frequency component at the p-th frequency bin, only N CMACs are required. Of course the fast Fourier transform (FFT) is usually used instead of the DFT, and this requires $N\log_2 N$ CMACs. Therefore if a Fourier transform is being performed simply to find if a tonal component is present at one frequency only, it makes more sense to use the DFT. Note that by the nature of the calculation data flow, the FFT cannot calculate a frequency component at one frequency only - it's all bins or none. *Goertzel's algorithm* provides a formal algorithmic procedure for calculating a single bin DFT.

Goertzel's algorithm to calculate the p-th frequency bin of an N point DFT is given by:

$$\begin{aligned} s_p(k) &= x(k) + 2\cos\left(\frac{2\pi p}{N}\right) s_p(k-1) - s_p(k-2) \\ y_p(k) &= s_p(k) - W_N^p s_p(k-1) \end{aligned} \tag{214}$$

where $W_N^p = e^{j\frac{2\pi p}{N}}$ and the initial conditions $s_p(-2) = s_p(-1) = 0$ apply.

Eq. 214 calculates the p-th frequency bin of the DFT after the algorithm has processed N data points, i.e. $X(p) = y_p(N)$. *Goertzel's algorithm* can be represented as a second order IIR:



An IIR filter representation of *Goertzel's algorithm*. Note that the non-recursive part of the filter has complex weights, whereas the recursive part has only real weights. The recursive part of this filter is in fact a simple narrowband filter. For an efficient implementation it is best to compute $s_p(k)$ for N samples, and thereafter evaluate $y_p(N)$.

For tone detection (i.e. tone present or not-present), only the signal power of the p-th frequency bin is of interest, i.e. $|X(p)|^2$. Therefore from Eq. 214:

$$\begin{aligned} |X(p)|^2 &= X(p)X^*(p) = y_p(N)y_p^*(N) \\ &= s_p(N)s_p(N) + 2\cos\left(\frac{2\pi p}{N}\right)s_p(N)s_p(N-1) + s_p(N-1)s_p(N-1) \end{aligned} \tag{215}$$

Goertzel's algorithm is widely used for dual tone multifrequency (DTMF) tone detection because of its simplicity and that it requires less computation than the DFT or FFT. For DTMF tones, there are 8 separate frequencies which must be detected. Therefore a total of 8 frequency bins are required. The International Telecommunication Union (ITU) suggest in standards Q.23 and Q24 that a 205 point DFT is performed for DTMF detection. To do a full DFT would require $205 \times 205 = 42025$ complex multiplies and adds (CMACs). To use a zero padded 256 point FFT would require $256\log_2 256 = 2048$ CMACs. Given that we are only interested in 8 frequency bins (and not 205 or 256), the computation required by *Goerztel's algorithm* is $8 \times 205 = 1640$ CMACs. Compared to the FFT, Goertzel's algorithm is simple and requires little memory or assembly language code to program. For DTMF tone detection the frequency bins corresponding to the second harmonic of each tone are also calculated. Hence the total computation of *Goertzel's algorithm* in this case is 3280 CMACs which is more than for the FFT. However the simplicity of *Goertzel's algorithm* means it is still the technique of choice.

In order to detect the tones at the DTMF frequencies, and using a 205 point DFT with $f_s = 8000$ Hz, the frequency bins to evaluate via *Geortzel's algorithm* are:

frequency, f/Hz	*bin*
697	18
770	20
852	22

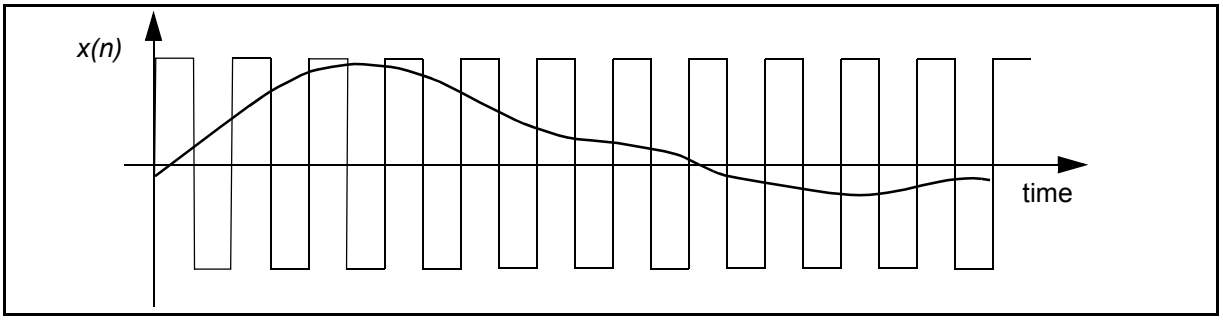
frequency, f / Hz	*bin*
941	24
1209	31
1336	34
1477	38
1633	42

Note that if the sampling frequency is not 8000 Hz, or a different number of data points are used, then the bin numbers will be different from above. See also *Discrete Fourier Transform, Dual Tone Multifrequency*, *Fast Fourier Transform*.

Gram-Schmidt: See *Matrix Decompositions - Gram-Schmidt.*

Granular Synthesis: A technique for musical instrument sound synthesis [13], [14], [32]. See also *Music, Western Music Scale*.

Granularity Effects: If the step size is too large in a delta modulator, then the delta modulated signal will give rise to a large error and completely fail to encode signals with a magnitude less than the step size. See also *Delta Modulation, Slope Overload.*



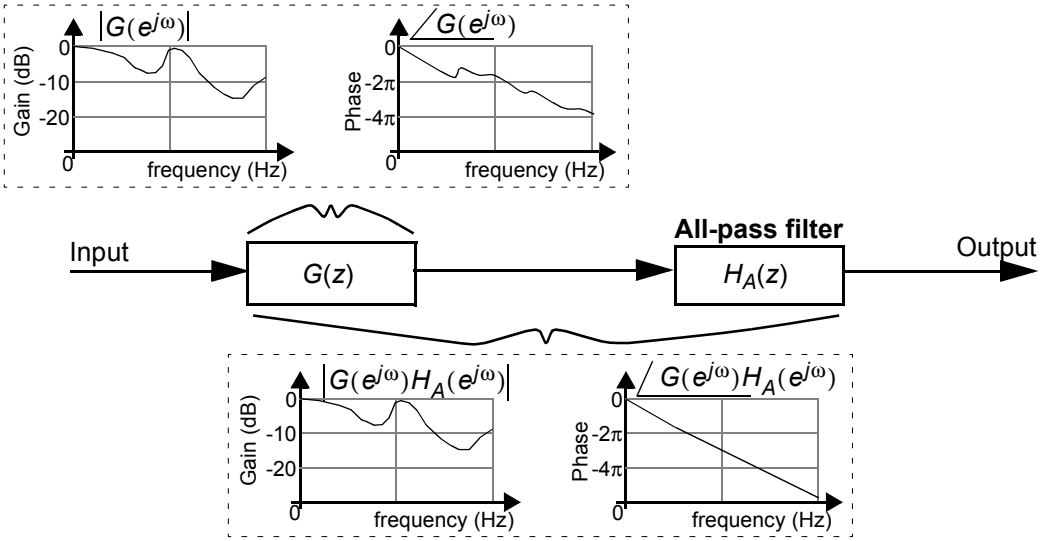
Graphic Interchange Format (GIF): The *GIF* format has become a de facto industry standard for the interchange of raster graphic data. *GIF* was first developed by Compuserve Inc, USA. *GIF* essentially defines a protocol for on-line transmission and interchange of raster graphic data such that it is completely independent of the hardware used to create or display the image. *GIF* has a limited, non-exclusive, royalty-free license and has widespread use on the Internet and in many DSP enabled multimedia systems. See also *Global Information Infrastructure, Joint Photographic Experts Group, Standards*.

Graphical Compiler: A system that allows you to *draw* your algorithm and application architecture on a computer screen using a library of icons (FIR filters, FFTs etc.) which will then be compiled into executable code, usually 'C', which can then be cross compiled to an appropriate assembly language for implementation on a DSP processor. See also *Cross Compiler.*

Graphical Equalizer: This is a device used in music systems which can be used to control the frequency content of the output. A *graphic equalizer* is therefore effectively a set of bandpass filters with independent gain settings that can be implemented in the analog or digital domains.

Group Delay: See *Finite Impulse Response Filter*.

Group Delay Equalisation: A technique to equalise the phase response of a system to be linear (i.e. constant group delay) by cascading the output of the system with an all pass filter designed to have suitable phase shifting characteristics. The magnitude frequency response of the system cascaded with the all pass filter is the same as that of the system on its own.



Group delay equalisation by cascading an all pass filter $H_A(z)$ with a non-linear phase filter $G(z)$ in order to linearise the phase response and therefore produce a constant group delay. The magnitude frequency response of the cascaded system, $|G(e^{j\omega})H_A(e^{j\omega})|$ is the same as the original system, $|G(e^{j\omega})|$ ..

The design of group delay equalisers is not a trivial procedure. See also *All-pass Filter, Equalisation, Finite Impulse Reponse Filter - Linear Phase* .

Group Speciale Mobile (GSM): The European mobile communication system that implements 13.5kbps speech coding (with half-rate 6.5kbps channels optional) and uses Gaussian Minimum Shift Keying (GMSK) modulation [85]. Data transmission is also available at rates slightly below the speech rates. See also *Minimum Shift Keying.*

H

H261: See *H-Series Recommendations - H261*.

H320: See *H-Series Recommendations - H320*.

H-Series Recommendations: The *H-series recommendations* from the International Telecommunication (ITU), advisory committee on telecommunications (denoted ITU-T, and formerly known as CCITT) propose a number of standards for the line transmission of non-telephone signals. Some of the current ITU-T *H-series recommendations* (http://www.itu.ch) can be summarised as:

H.100	Visual telephone systems.
H.110	Hypothetical reference connections for videoconferencing using primary digital group transmission.
H.120	Codecs for videoconferencing using primary digital group transmission.
H.130	Frame structures for use in the international interconnection of digital codecs for videoconferencing or visual telephony
H.140	A multipoint international videoconference system
H.200	Framework for Recommendations for audiovisual services
H.221	Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices
H.224	A real time control protocol for simplex application using the H.221 LSD/HSD/MLP channels.
H.230	Frame-synchronous control and indication signals for audiovisual systems.
H.231	Multipoint control units for audiovisual systems using digital channels up to 2 Mbit/s.
H.233	Confidentiality system for audiovisual services.
H.234	Encryption key management and authentication system for audiovisual services.
H.242	System for establishing communication between audiovisual terminals using digital channels up to 2 Mbit/s.
H.243	Procedures for establishing communication between three or more audiovisual terminals using digital channels up to 2 Mbit/s.
H.261	Video codec for audiovisual services at p x 64 kbit/s.
H.281	A far end camera control protocol for videoconferences using H.224.
H.320	Narrow-band visual telephone systems and terminal equipment below.
H.331	Broadcasting type audiovisual multipoint systems and terminal equipment.

From the interest point of DSP and multimedia systems and algorithms the above title descriptions of H242, H261 and H320 can be expanded upon as per http://www.itu.ch:

- **H.242:** The H242 recommendation defines audiovisual communication using digital channels up to 2 Mbit/s. This recommendation should be read in conjunction with ITU-T recommendations G.725, H.221 and H.230. H242 is suitable for applications that can use narrow (3 kHz) and wideband (7 kHz) speech together with video such as video-telephony, audio and videoconferencing and so on. H242 can produce speech, and optionally video and/ or data at several rates, in a number of different modes. Some applications will require only a single channel, whereas others may require two or more channels to provide the higher bandwidth.
- **H.261:** The H.261 recommendation describes video coding and decoding methods for the moving picture component of audiovisual services at the rate of p x 64 kbit/s, where p is an integer in the range 1 to 30, i.e. 64kbits/s to 1.92Mbits/s. H261 is suitable for transmission of video over ISDN lines, for applications such as videophones and videoconferencing. The videophone application can tolerate a low image quality and can be achieved for $p = 1$ or 2. For videoconferencing applications where the transmission image is likely to include a few people and last for a long period, higher picture quality is required and $p > 6$ is required. H.261 defines two picture formats: CIF (Common Intermediate Format) has 288 lines by 360 pixels/line of luminance information and 144 x 180 of chrominance information; and QCIF (Quarter Common Intermediate Format) which is 144 lines by 180 pixels/line of luminance and 72 x 90 of chrominance. The choice of CIF or QCIF depends on available channel capacity and desired quality.

The H261 encoding algorithm is similar in structure that of MPEG, however they are not compatible. It is also worth noting that H.261 requires considerably less CPU power for encoding than MPEG. Also the algorithm makes available use of the bandwidth by trading picture quality against motion. Therefore a fast moving image will have a lower quality than a static image. H.261 used in this way is thus a constant-bit-rate encoding rather than a constant-quality, variable-bit-rate encoding.

- H.320: H.320 specifies a narrow-band visual telephone services for use in channels where the data rate cannot exceed 1920 kbit/s.

For additional detail consult the appropriate standard document or contact the ITU. See also *International Telecommunication Union, ITU-T Recommendations, Standards*.

Haas Effect: In a reverberant environment the sound energy received by the direct path can be much lower than the energy received by indirect reflective paths. However the human ear is still able to localize the sound location correctly by localizing the first components of the signal to arrive. Later echoes arriving at the ear increase the perceived loudness of the sound as they will have the same general spectrum. This psychoacoustic effect is commonly known as the precedence effect, the law of the first wavefront, or sometimes the *Haas effect* [30]. The *Haas effect* applies mainly to short duration sounds or those of a discontinuous or varying form. See also *Ear, Lateralization, Source Localization, Threshold of Hearing.*

Habituation: *Habituation* is the effect of the auditory mechanism not perceiving a repetitive noise (which is above the threshold of hearing) such as the ticking of a nearby clock or passing of nearby traffic until attention is directed towards the sound. See also *Adaptation, Psychoacoustics, Threshold of Hearing.*

Hamming Distance: Often used in channel coding applications, *Hamming distance* refers to the number of bit locations in which two binary codewords differ. For example the binary words *10100011* and *10001011* differ in two positions (the third and the fifth from the left) so the Hamming distance between these words is *2*. See also *Euclidean Distance, Channel Coding, Viterbi Algorithm.*

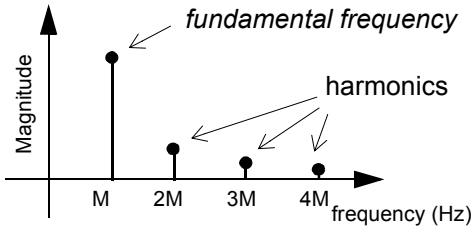
Hamming Window: See *Windows.*

Half Duplex: Pertaining to the capability to send and receive data on the same line, but not simultaneously. See also *Full Duplex, Simplex*.

Hand Coding: When writing programs for DSP processors 'C' cross compilers are often available. Although algorithm development with cross compilers is faster than when using assembly language, the machine code produced is usually less efficient and compact as would be achieved by writing in assembler. Cleaning up this less efficient assembly code is sometimes referred to as *hand-coding*. Coding directly in machine code is also referred to as *hand-coding*. See also *Assembly Language, Cross-Compiler, Machine Code.*

Handshaking: A communication technique whereby one system acknowledges receipt of data from another system by sending a *handshaking* signal.

Harmonic: Given a signal with fundamental frequency of *M* Hz, *harmonics* of this signal are at integer multiples of *M*, i.e. at 2*M*, 3*M*, 4*M*, and so on. See also *Fundamental Frequency, Music, Sub-harmonic, Total Harmonic Distortion*.



The frequency domain representation of a tone at M Hz with associated harmonics.

harris Window: See *Windows.*

Hartley Transform: The *Hartley transform* is "similar" in computational structure (although different in properties) to the Fourier transform. One key difference is that the *Hartley transform* uses real numbers rather than complex numbers. A good overview of the mathematics and application of the *Hartley transform* can be found in [121].

Harvard Architecture: A type of microprocessor (and microcomputer) architecture where the memory used to store the program, and the memory used to store the data are separate therefore allowing both program and data to be accessed simultaneously. Some DSPs are described as being a modified *Harvard architecture* where both program and data memories are separate, but with cross-over links. See also *DSP Processor*.

Head Shadow: Due to the shape of the human head, incident sounds can be diffracted before reaching the ears. Hence the actual waveform arriving at the ears is different than what would have been received by an ear without the head present. *Headshadow* is an important consideration in the design of virtual sound systems and in the design of some types of advanced DSP hearing aids. See also *Diffraction.*

Hearing: The mechanism and process by which mammals perceive changes in acoustic pressure waves, or sound. See also *Audiology, Audiometry, Ear, Psychoacoustics, Threshold of Hearing*.

Hearing Aids: A *hearing aid* can be described as any device which aids the wearer by improving the audibility of speech and other sounds. The simplest form of *hearing aid* is an acoustic amplification device (such as an ear trumpet), and the most complex is probably a cochlear implant system (surgically inserted) which electrically stimulates nerves using acoustic derived signals received from a body worn radio transmitter and microphone.

More commonly, *hearing aids* are recognizable as analogue electronic amplification devices consisting of a microphone and amplifier connected to an acoustic transducer usually just inside the ear. However a *hearing aid* which simply makes sounds louder is not all that is necessary to allow hearing impaired individuals to hear better. In everyday life we are exposed to a very wide range of sounds coming from all directions with varying intensities, and various degrees of reverberation. Clearly *hearing aids* are required to be very versatile instruments, that are carefully designed around known parameters and functions of the ear, and providing compensation techniques that are suitable for the particular type of hearing loss, in particular acoustic environments.

Simple analogue electronic *hearing aids* can typically provide functions of volume and tone control. More advanced devices may incorporate multi-band control (i.e., simple frequency shaping) and automatic gain control amplifiers to adjust the amplification when loud noises are present. *Hearing aids* offering multi-band compression with a plethora of digitally adjustable parameters such as attack and release times, etc., have become more popular. Acoustic feedback reduction techniques have also been employed to allow more amplification to be provided before the microphone/ transducer loop goes unstable due to feedback (this instability is often detected as an unsatisfied hearing aid wearer with a screeching howl in their ear). Acoustic noise reduction aids that exploit the processing power of advanced DSP processing have also been designed.

Digital audio signal processing based *hearing aids* may have advantages over traditional analogue audio *hearing aids*. They provide a greater accuracy and flexibility in the choice of electroacoustic parameters and can be easily interfaced to a computer based audiometer. More importantly they can use powerful adaptive signal processing techniques for enhancing speech intelligibility and reducing the effects of background noise and reverberation. Currently however, power and physical size constraints are limiting the availability of DSP *hearing aids*. See also *Audiology, Audiometry, Beamforming, Ear, Head Shadow, Hearing Impairment, Threshold of Hearing.*

Hearing Impairment: A reduction in the ability to perceive sound, as compared to the average capability of a cross section of unimpaired young persons. *Hearing impairment* can be caused by exposure to high sound pressure levels (SPL), drug induced, virus-induced, or simply as a result of having lived a long time. A hearing loss can be simply quantified by an audiogram and qualified with more exact audiological language such as sensorineural loss or conductive loss, etc., [4], [30]. See also *Audiology, Audiometry, Conductive Hearing Loss, Ear, Hearing, Loudness Recruitment, Sensorineural Hearing Loss, Sound Pressure Level, Threshold of Hearing.*

Hearing Level (HL): When the hearing of person is to be tested, the simplest method is to play pure tones through headphones (using a calibrated audiometer) over a range of frequencies, and determine the minimum sound pressure level (SPL) at which the person can hear the tone. The results could then be plotted as minimum perceived SPL versus frequency. To ascertain if the person has a hearing impairment the plot can be compared with the average minimum level of SPL for a cross section of healthy young people with no known hearing impairments. However if the minimum level of SPL (the threshold of hearing) is plotted as SPL versus frequency, the curve obtained is not a straight line and comparison can be awkward. Therefore for *Hearing Level (dB)* plots (or audiograms), the deviation from the average threshold of hearing of young people is plotted with hearing loss indicated by a positive measurement that is plotted *lower* on the audiogram. The threshold of hearing is therefore the 0dB line on the *Hearing Level (dB) scale*. The equivalent dB (HL) and dB (SPL) for some key audiometric frequencies in the UK are [157]:

Frequency (Hz)	250	500	1000	2000	4000	8000
dB (HL)	0	0	0	0	0	0
dB (SPL)	26	15.6	8.2	5.2	7	20

See also *Audiogram, Audiometry, Equal Loudness Contours, Frequency Range of Hearing, Hearing Impairment, Loudness Recruitment, Sensation Level, Sound Pressure Level, Threshold of Hearing*.

Hearing Loss: See *Hearing Impairment.*

Hermitian: See *Matrix Properties - Hermitian Transpose.*

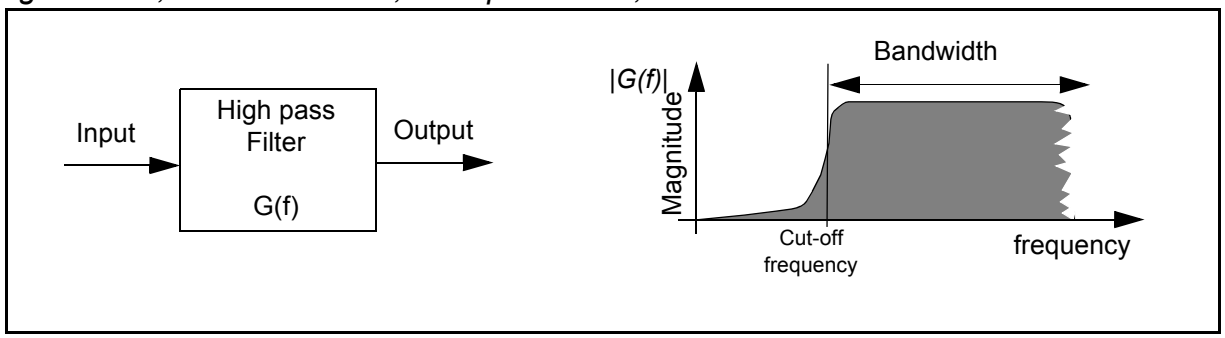
Hermitian Transpose: See *Matrix Properties - Hermitian Transpose*.

Hertz (Hz): The unit of frequency measurement named after Heinrich *Hertz*. 1 Hz is 1 cycle per second.

Hexadecimal, Hex: Base 16. Conversion from binary to hex is very straightforward and therefore *hex* digits have become the standard way of representing binary quantities to programmers. A 16 bit binary number can be easily represented in 4 *hex* digits by grouping four bits together starting from the binary point and converting to the corresponding *hex* digit. The *hex* digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Hexadecimal entries in DSP assembly language programs are prefixed by either by $ or *0x* to differentiate them from decimal entries. An example (with base indicated as subscript):

$0010\ 1010\ 0011\ 1111_2 = 2A3F_{16} = (2 \times 16^3) + (10 \times 16^2) + (3 \times 16^1) + 15 = 10815_{10}$

High Pass Filter: A filter which passes only the portions of a signal that have frequencies above a specified cut-off frequency. Frequencies below the cut-off frequency are highly attenuated. See also *Digital Filter, Low Pass Filter, Bandpass Filter, Filters*.



Higher Order Statistics: Most stochastic DSP techniques such as the power spectrum, least mean squares algorithm and so on, are based on first and second order statistical measures such as mean, variance and autocorrelation. The higher order moments, such as the 3rd order moment (note that the first order moment is the mean, the second order central moment is the variance) are usually not considered. However there is information to be gathered from a consideration of these *higher order statistics*. One example is detecting the baud rate of PSK signals. Recently there has been considerable interest in higher order statistics within the DSP community. For information refer to the tutorial article [117]. See also *Mean, Variance*.

Hilbert Transform: Simply described, a Hilbert transform introduces a phase shift of 90 degrees at all frequencies for a given signal. A Hilbert transform can be implemented by an all-pass phase shift network. Mathematically, the Hilbert transform of a signal $x(t)$ can be computed by linear filtering (i.e., convolution) with a special function:

$$x_h(t) \equiv x(t) \otimes \frac{1}{\pi t} \tag{216}$$

It may be more helpful to think about the Hilbert transform as a filtered version of a signal rather than a “transform” of a signal. The Hilbert transform is useful in constructing single sideband signals (thus conserving bandwidth in communications examples). The transform is also useful in signal analysis by allowing real bandpass signals (such as a radio signal) to be analyzed and simulated

as an equivalent complex baseband (or lowpass) process. Virtually all system simulation packages exploit this equivalent representation to allow for timely completion of system simulations. Not obvious from the definition above is the fact that the Hilbert transform of the Hilbert transform of $x(t)$ is $-x(t)$. This may be expected from the heuristic description of the Hilbert transform as a 90 degree phase shift -- i.e., two 90 degree phase shifts are a 180 degree phase shift which means multiplying by a minus one.

Host: Most DSP boards can be *hosted* by a general purpose computer, such as an IBM compatible PC. The host allows a DSP designer to develop code using the PC, and then download the DSP program to the DSP board. The DSP board therefore has a *host interface*. The host usually supplies power (analog, 12V and digital, 5V) to the board. See also *DSP Board*.

Householder Transformation: See *Matrix Decompositions - Householder Transformation*.

Huffman Coding: This type of coding exploits the fact that discrete amplitudes of a quantized signal may not occur with equal probability. Variable length codewords can therefore be assigned to a particular data sequence according to their frequency of occurrence. Data that occurs frequently are assigned shorter code words, hence data compression is possible.

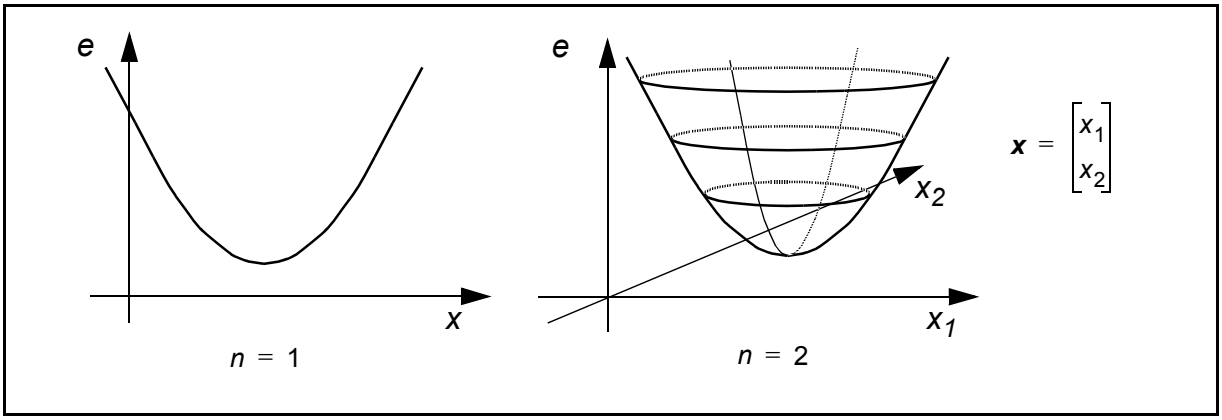
Hydrophone: An underwater transducer of acoustic energy for sonar applications.

Hyperchief: A MacIntosh program developed by a DSP graduate student from 1986 - 1991, somewhere on the west coast of the USA, to simulate the wisdom of a Ph.D. supervisor. However, while accurately simulating the wisdom of a Ph.D. supervisor, *Hyperchief* precisely illustrated the pitfalls of easy access to powerful computers. *Hyperchief* is sometime spelled as Hypercheif (pronounced Hi-per-chife).

Hyperparaboloid: Consider the equation:

$$e = \boldsymbol{x}^T \boldsymbol{R} \boldsymbol{x} + 2\boldsymbol{p}^T \boldsymbol{x} + s \tag{217}$$

where $\boldsymbol{x}$ is an $n \times 1$ vector, $\boldsymbol{R}$ is a positive definite $n \times n$ matrix, $\boldsymbol{p}$ is an $n \times 1$ vector, and s is a scalar. The equation is quadratic in $\boldsymbol{x}$. If $n = 1$, then e will form a simple parabola, and if $n = 2$, e can be represented as a (solid) paraboloid:



The positive definiteness of $\boldsymbol{R}$ ensures that the parabola is up-facing. Note that in both cases the e has exactly one minimum point (a global minimum) at the bottom of the parabolic shape. For systems with $n \geq 3$ e cannot be shown diagrammatically as four or more dimensions are required! Hence we are asked to imagine the existence of a *hyperparaboloid* for $n \geq 3$ and which will also have exactly one minimum point for e. The existence of the *hyperparaboloid* is much referred to for

least squares, and least mean squares algorithm derivations. See also *Global Minimum, Local Minima.*

Hypersignal: An IBM PC based program for DSP written by Hyperception Inc. *Hypersignal* provides facilities for real time data acquisition in conjunction with various DSP processors, and a menu driven system to perform off-line processing of real-time FFTs, digital filtering, signal acquisition, signal generation, power spectra and so on. DOS and Windows versions are available.

HyTime: *HyTime* (Hypermedia/Time-Based Structuring Language) is a standardised infrastructure for the representation of integrated, open hypermedia documents produced by the International Organization for Standards (ISO), Joint Technical Committee, Sub Committee (SC) 18, Working Group (WG) 8 (ISO JTC1/SC18/WG8). See also *Bento, Multimedia and Hypermedia Information Coding Experts Group, Standards*.

I

i: "*i*" (along with "k" and "n") is often used as a discrete time index for in DSP notation. See *Discrete Time*.

I: Often used to denoted the identity matrix. See *Matrix*.

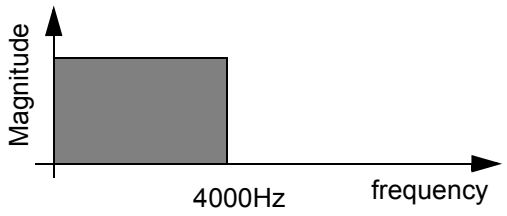
I-Series Recommendations: The I-series telecommunication recommendations from the International Telecommunication (ITU), advisory committee on telecommunications (denoted ITU-T and formerly known as CCITT) provide standards for Integrated Services Digital Networks. Some of the current recommendations (http://www.itu.ch) include:

- I.112 Vocabulary of terms for ISDNs.
- I.113 Vocabulary of terms for broadband aspects of ISDN.
- I.114 Vocabulary of terms for universal personal telecommunication.
- I.120 Integrated services digital networks (ISDNs).
- I.121 Broadband aspects of ISDN.
- I.122 Framework for frame mode bearer services.
- I.140 Attribute technique for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN.
- I.141 ISDN network charging capabilities attributes.
- I.150 B-ISDN asynchronous transfer mode functional characteristics.
- I.200 Guidance to the I.200-series of Recommendations.
- I.210 Principles of telecommunication services supported by an ISDN and the means to describe them.
- I.211 B-ISDN service aspects.
- I.220 Common dynamic description of basic telecommunication services.
- I.221 Common specific characteristics of services.
- I.230 Definition of bearer service categories.
- I.231 Circuit-mode bearer service categories.
- I.231.9 Circuit mode 64 kbit/s 8 kHz structured multi-use bearer service category.
- I.231.10 Circuit-mode multiple-rate unrestricted 8 kHz structured bearer service category.
- I.232 Packet-mode bearer services categories.
- I.232.3 User signalling bearer service category (USBS).
- I.233 Frame mode bearer services.
- I.233.1-2 ISDN frame relaying bearer service/ ISDN frame switching bearer service.
- I.241.7 Telephony 7 kHz teleservice.
- I.250 Definition of supplementary services.
- I.251.1-9 Direct-dialling-in/ Multiple subscriber number/ Calling line identification presentation/ Calling line identification restriction/ Connected Line Identification Presentation (COLP)/ Connected Line Identification Restriction (COLR)/ Malicious call identification/ Sub-addressing supplementary service.
- I.252.2-5 Call forwarding busy/ Call forwarding no reply/ Call forwarding unconditional/ Call deflection.
- I.253.1-2 Call waiting (CW) supplementary service/ Call hold.
- I.254.2 Three-party supplementary service.
- I.255.1 Closed user group.
- I.255.3-5 Multi-level precedence and preemption service (MLPP)/ Priority service/ Outgoing call barring.
- I.256 Advice of charge
- I.257.1 User-to-user signalling.
- I.258.2 In-call modification (IM).
- I.310 ISDN Network functional principles.
- I.311 B-ISDN general network aspects.
- I.312 (See also Q.1201.) Principles of intelligent network architecture.
- I.320 ISDN protocol reference model.
- I.321 B-ISDN protocol reference model and its application.
- I.324 ISDN network architecture.

I.325 Reference configurations for ISDN connection types.
I.327 B-ISDN functional architecture.
I.328 Intelligent Network - Service plane architecture.
I.329 Intelligent Network - Global functional plane architecture.
I.330 ISDN numbering and addressing principles.
I.331 Numbering plan for the ISDN era.
I.333 Terminal selection in ISDN.
I.334 Principles relating ISDN numbers/subaddresses to the OSI reference model network layer addresses.
I.350 General aspects of quality of service and network performance in digital networks, including ISDNs.
I.351 Relationships among ISDN performance recommendations.
I.352 Network performance objectives for connection processing delays in an ISDN.
I.353 Reference events for defining ISDN performance parameters.
I.354 Network performance objectives for packet mode communication in an ISDN.
I.355 ISDN 64 kbit/s connection type availability performance.
I.356 B-ISDN ATM layer cell transfer performance.
I.361 B-ISDN ATM layer specification.
I.362 B-ISDN ATM Adaptation Layer (AAL) functional description.
I.363 B-ISDN ATM adaptation layer (AAL) specification.
I.364 Support of broadband connectioneless data service on B-ISDN.
I.365.1 Frame relaying service specific convergence sublayer (FR-SSCS).
I.370 Congestion management for the ISDN frame relaying bearer service.
I.371 Traffic control and congestion control in B-ISDN.
I.372 Frame relaying bearer service network-to-network interface requirements.
I.373 Network capabilities to support Universal Personal Telecommunication (UPT).
I.374 Framework Recommendation on “Network capabilities to support multimedia services”.
I.376 ISDN network capabilities for the support of the teleaction service.
I.410 General aspects and principles relating to Recommendations on ISDN user-network interfaces.
I.411 ISDN user-network interfaces - references configurations.
I.412 ISDN user-network interfaces - Interface structures and access capabilities.
I.413 B-ISDN user-network interface.
I.414 Overview of Recommendations on layer 1 for ISDN and B-ISDN customer accesses.
I.420 Basic user-network interface.
I.421 Primary rate user-network interface.
I.430 Basic user-network interface - Layer 1 specification.
I.431 Primary rate user-network interface - Layer 1 specification.
I.432 B-ISDN user-network interface - Physical layer specification.
I.460 Multiplexing, rate adaption and support of existing interfaces.
I.464 Multiplexing, rate adaption and support of Existing interfaces for restricted 64 kbit/s transfer capability.
I.470 Relationship of terminal functions to ISDN.
I.500 General structure of the ISDN interworking Recommendations.
I.501 Service interworking.
I.510 Definitions and general principles for ISDN interworking.
I.511 ISDN-to-ISDN layer 1 internetwork interface.
I.515 Parameter exchange for ISDN interworking.
I.520 General arrangements for network interworking between ISDNs.
I.525 Interworking between ISDN and networks which operate at bit rates of less than 64 kbit/s.
I.530 Network interworking between an ISDN and a public switched telephone network (PSTN).
I.555 Frame relaying bearer service interworking.
I.570 Public/private ISDN interworking.
I.580 General arrangements for interworking between B-ISDN and 64 kbit/s based ISDN.
I.601 General maintenance principles of ISDN subscriber access and subscriber installation.
I.610 B-ISDN operation and maintenance principles and functions.

For additional detail consult the appropriate standard document or contact the ITU. See also *International Telecommunication Union, ITU-T Recommendations, Standards*.

Ideal Filter: The *ideal filter* for a DSP application is one which will give absolute discrimination between passband and stopband. The impulse response of an *ideal filter* is always non-causal, and therefore impossible to build. See also *Brick Wall Filter, Digital Filter* .



A brick wall filter cutting off at 4000Hz is the ideal anti-alias filter for a DSP application with f_s = 8000Hz. All frequencies below 4000Hz are passed perfectly with no amplitude or phase distortion, and all frequencies above 4000Hz are removed. In practice the ideal filter cannot be achieved as it would be non-causal. In an FIR implementation, the more weights that are used, the closer the frequency response will be to the ideal.

Identity Matrix: See *Matrix Structured - Identity.*

IEEE 488 GPIB: Many DSP laboratory instruments such as data loggers and digital oscilloscopes are equipped with a *GPIB* (General Purpose Interface Bus). Note that this bus is also referred to as HPIB by Hewlett-Packard, developers of the original bus on which the standard is based. Different devices can then communicate through cables of maximum length 20 metres using an 8-bit parallel protocol with a maximum data transfer of 2Mbytes/sec.

IEEE Standard 754: The IEEE Standard for binary floating point arithmetic specifies basic and extended floating-point number formats; add, subtract, multiply, divide, remainder, and square root. It also provides magnitude compare operations, conversion from/to integer and floating-point formats and conversions between different floating-point formats and decimal strings. Finally the standard also specifies floating-point exceptions and their handling, including non-numbers caused by divide by zero. The Motorola DSP96000 is an IEEE 754 compliant floating point processor. Devices such as the Texas Instruments TMS320C30 use their similar (but different!) floating point format. The IEEE Standard 754 has also been adopted by ANSI and is therefore often referred to as ANSI/IEEE Standard 754. See also *Standards*.

IEEE Standards: The IEEE publish standards in virtually every conceivable area of electronic and electrical engineering. These standards are available from the IEEE and the titles, classifications and a brief synopsis can be browsed at http://stdsbbs.ieee.org. See also *Standards.*

Ill-Conditioned: See *Matrix Properties - Ill-Conditioned.*

Image Interchange Facility (IIF): The *IIF* has been produced by the International Organization for Standards (ISO,) Joint Technical Committee (JTC) 1, sub-committee (SC) 24 (ISO/IEC JTC1/ SC24) which is responsible for standards on “Computer graphics and image processing”. The *IIF* standard is ISO 12087-3 and is the definition of a data format for exchanging image data of an arbitrary structure. The *IIF* format is designed to allow easy integration into international telecommunication services. See also *International Organisation for Standards, JBIG, JPEG, Standards*.

Imaginary Number: The *imaginary number* denoted by *j* for electrical engineers (and by most other branches of science and mathematics) is the square root of -1. Using *imaginary numbers*

allows the square root of any negative number to be expressed. For example, $\sqrt{-25} = 5j$. See also *Complex Numbers, Fourier Analysis, Euler's Formula*.

Impulse: An *impulse* is a signal with very large magnitude which lasts only for a very short time. A mechanical impulse could be applied by striking an object with a hammer; a very large force for a very short time. A voltage *impulse* would be a very large voltage signal which only lasts for a few milli- or even microseconds.

A digital *impulse* has magnitude of 1 for one sample, then zero at all other times and is sometimes called the unit impulse or unit pulse. The mathematical notation for an *impulse* is usually $\delta(t)$ for an analog signal, and $\delta(n)$ for a digital impulse. For more details see *Unit Impulse Function,*. See also *Convolution, Elementary Signals, Fourier Transform Properties, Impulse Response, Sampling Property, Unit Impulse Function, Unit Step Function.*

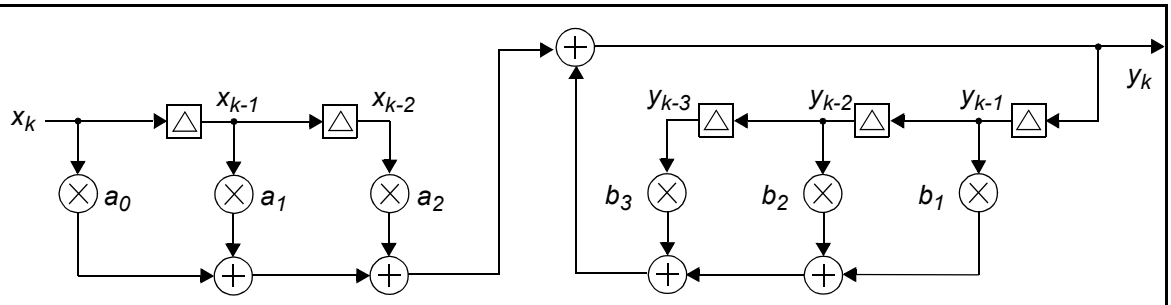
Impulse Response: When any system is excited by an impulse, the resulting output can be described as the *impulse response* (or the response of the system to an impulse). For example, striking a bell with a hammer gives rise to the familiar ringing sound of the bell which gradually decays away. This ringing can be thought of as the bell's *impulse response*, which is characterized by a slowly decaying signal at a fundamental frequency plus harmonics. The bell's physical structure supports certain modes of vibrations and suppresses others. The impulsive input has energy at all frequencies -- the frequencies associated with the supported modes of vibration are sustained while all other frequencies are suppressed. These sustained vibrations gives rise to the bell's ringing sound that we hear (after the extremely brief "chink" of the impulsive hammer blow).

We can also realize the digital *impulse response* of a system by applying a unit impulse and observing the output samples that result. From the *impulse response* of any linear system we can calculate the output signal for any given input signal simply by calculating the convolution of the *impulse response* with the input signal. Taking the Fourier transform of the *impulse response* of a system gives the frequency response. See also *Convolution, Elementary Signals, Fourier Transform Properties, Impulse, Sampling Property, Unit Impulse Function, Unit Step Function.*

Incoherent: See *Coherent.*

Infinite Impulse Response (IIR) Filter: A digital filter which employs feedback to allow sharper frequency responses to be obtained for fewer filter coefficients. Unlike FIR filters, *IIR filters* can exhibit instability and must therefore be very carefully designed [10], [42]. The term infinite refers to the fact that the output from a unit pulse input will exhibit nonzero outputs for an arbitrarily long time.

If the digital filter is IIR, then two weight vectors can be defined: one for the feedforward weights and one for the feedback weights:



Feedforward Zeroes (non-recursive) *Feedback Poles (recursive)*

$$y_k = \sum_{n=0}^{2} a_n x_{k-n} + \sum_{n=1}^{3} b_n y_{k-n} = a_0 x_k + a_1 x_{k-1} + a_2 x_{k-2} + b_1 y_{k-1} + b_2 y_{k-2} + b_3 y_{k-3}$$

$$\Rightarrow y_k = \boldsymbol{a}^T \boldsymbol{x}_k + \boldsymbol{b}^T \boldsymbol{y}_{k-1} = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \\ x_{k-2} \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ y_{k-3} \end{bmatrix}$$

A signal flow graph and equation for a 3 zero, 4 pole infinite impulse response filter.

See also *Digital Filter, Finite Impulse Response Filter*, *Least Mean Squares IIR Algorithms*.

Infinite Impulse Response (IIR) LMS: See *Least Mean Squares IIR Algorithms.*

Infinity (∞) Norm: See *Matrix Properties - ∞ Norm.*

Information Theory: The name given to the general study of the coding of information. In 1948 Claude E. Shannon presented a mathematical theory describing, among other things, the average amount of information, or the entropy of a information source. For example, a given alphabet is composed of *N* symbols ($s_1, s_2, s_3, s_4, ..., s_N$). Symbols from a source that generates random elements from this alphabet are encoded and transmitted via a communication line. The symbols are decoded at the other end. Shannon described a useful relationship between information and the probability distribution of the source symbols: if the probability of receiving a particular symbol is very high then it does not convey a great deal of information, and if low, then it does convey a high degree of information. In addition, his measure was logarithmically based. According to Shannon's measure, the self information conveyed by a single symbol that occurs with probability P_i is:

$$I(s_i) = \log_2\left(\frac{1}{P_i}\right) \tag{218}$$

The average amount of information, or first order entropy, of a source can then be expressed as:

$$H_r(s) = \sum_{i=1}^{N} P_i \log_2\left(\frac{1}{P_i}\right) \tag{219}$$

Infrasonic: Of, or relating to infrasound. See *Infrasound*.

Infrasound: Acoustics signals (speed in air, $330ms^{-1}$) having frequencies below 20Hz, the low frequency limit of human hearing, are known as infrasound. Although sounds as low as 3Hz have been shown to be aurally detectable, there is no perceptible reduction in pitch and the sounds will also be tactile. Infrasound is a topic close to the heart of a number of professional recording engineers who believe that it is vitally important to the overall sound of music. In general CDs and DATs can record down to around 5Hz.

Exposure to very high levels *infrasound* can be extremely dangerous and certain frequencies can set cause organs and other body parts to resonate::

Area of Body	Approximate Resonance Range (Hz)
Motion sickness	0.3-0.6
Abdomen	3-5
Spine/pelvis	4-6
Testicle/Bladder	10
Head/Shoulders	20-30
Eyeball	60-90
Jaw/Skull	120-200

Infrasound has been considered as a weapon for the military and also as a means of crowd control, whereby the bladder is irritated. See also *Sound, Ultrasound*.

Inner Product: See *Vector Operations - Inner Product.*

In-Phase: See *Quadrature.*

Instability: A system or algorithm goes unstable when feedback (either physical or mathematical) causes the system output to oscillate uncontrollably. For example if a microphone is connected to an amplifier then to a loudspeaker, and the microphone is brought close to the speaker then the familiar feedback howl occurs; this is instability. Similarly in a DSP algorithm mathematical feedback in equations being implemented (recursion) may cause *instability*. Therefore to ensure a system is stable, feedback must be carefully controlled.

Institute of Electrical Engineers (IEE): The IEE is a UK based professional body representing electronic and electrical engineers The IEE publish a number of signal processing related publications each month, and also organize DSP related colloquia and conferences.

Institute of Electrical and Electronic Engineers, Inc. (IEEE): The IEEE is a USA based professional body covering every aspect of electronic and electrical engineering. IEEE publishes a very large number of journals each month which include a number of notable signal processing journals such Transactions on Signal Processing, Transactions on Speech and Audio Processing, Transactions on Biomedical Engineering, Transactions on Image Processing and so on.

Integration (1): The simplest mathematical interpretation of integration is taking the area under a graph.

Integration (2): The generic term for the implementation of many transistors on a single substrate of silicon. The technology refers to the actual process used to produce the transistors: CMOS is the integration technology for MOSFET transistors; Bipolar is the *integration* technology for TTL. The number of transistors on a single device is often indicated by one of the acronyms, SSI, MSI, LSI, VLSI, or ULSI.

Acronym	Technology	No. of Transistors	First Circuits	Example
SSI	Small scale integration	< 10	1960s	NPN junction
MSI	Medium Scale Integration	< 1000	1970s	4 NAND gates
LSI	Large Scale Integration	< 10000	Early 1980s	8086 microprocessor
VLSI	Very Large Scale Integration	<1000000	Mid 1980s	DSP56000
ULSI	Ultra Large Scale Integration	<100000000	1990s	TMS320C80

Integrated Circuit (IC): The name given to a single silicon chip containing many transistors that collectively realize some system level component such as an A/D converter or microprocessor.

Integrated Digital Services Network (ISDN): See *I-Series Recommendations.*

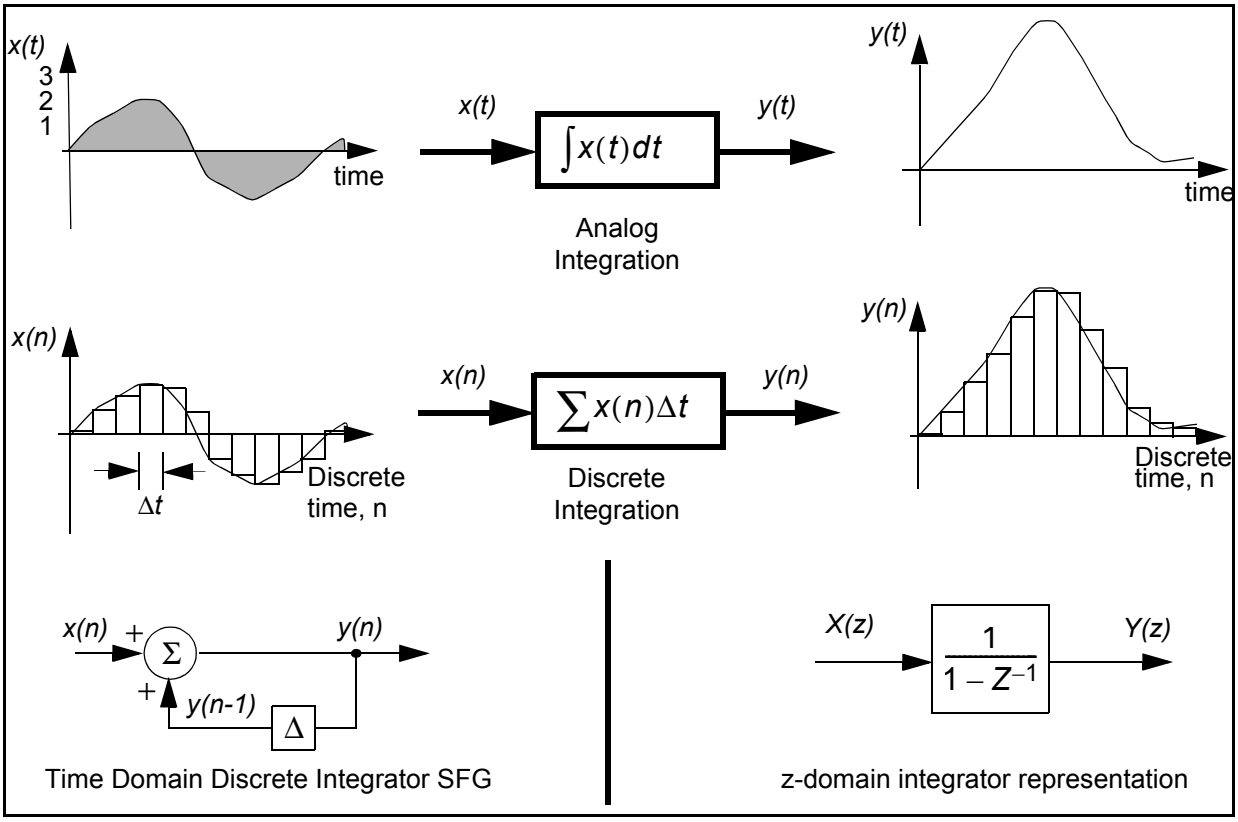
Integrator: A device which will performs the function of computing the integral as an output for an arbitrary input signal. In digital signal processing terms an *integrator* is quite straightforward. Consider the simple mathematical definition of integration which is the area under a graph. The output of an integrator, y(t), will be the area cumulative area under the input signal curve, x(t). For sampled digital signals the input will be constant for one sampling period, and therefore to approximately integrate the signal we can simply add the area of the sampling rectangles together. If the sampling period is normalized to one, then a signal can be integrated in the discrete domain by adding together the input samples. An *integrator* is implemented using a digital delay element, and a summing element which calculates the function:

$$y(n) = x(n) + y(n-1) \tag{220}$$

In the z-domain the transfer function of a discrete integrator is:

$$\begin{aligned} Y(z) &= X(z) + z^{-1}Y(z) \\ \Rightarrow \quad \frac{Y(z)}{X(z)} &= \frac{z}{z-1} \end{aligned} \tag{221}$$

When viewed in the frequency domain an integrator has the characteristics of a simple low pass filter. See also *Differentiator, Low Pass Filter*.

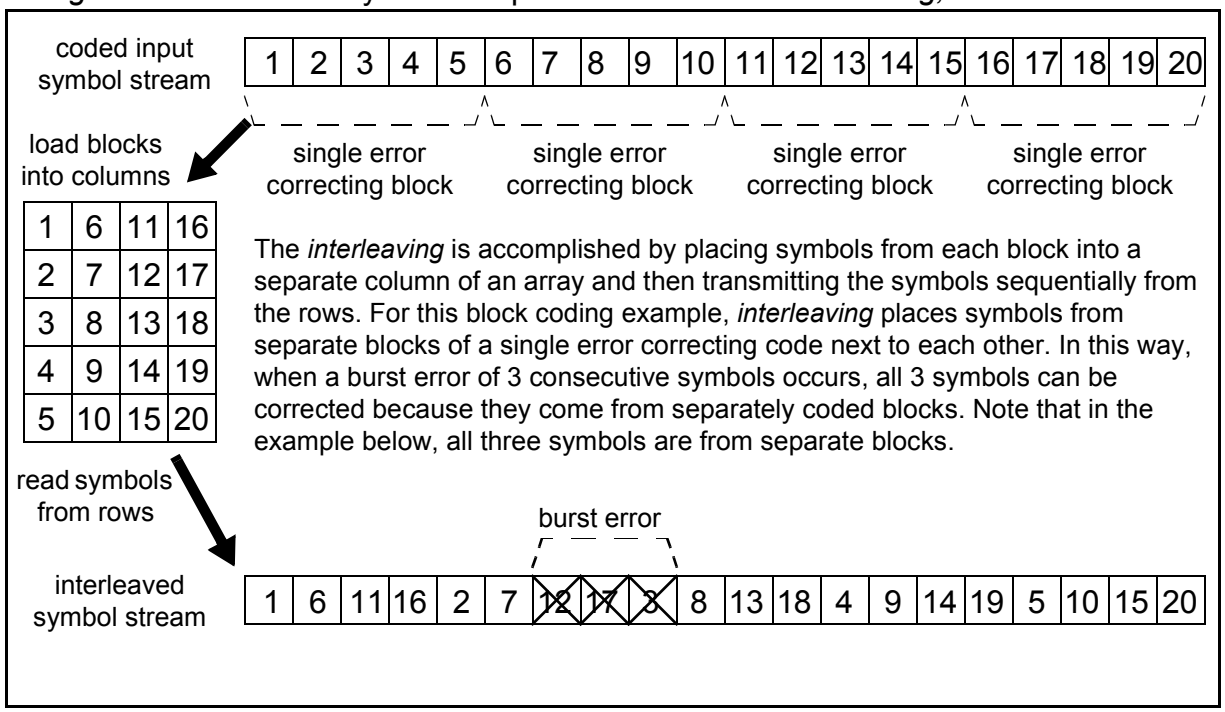


Intensity: See *Sound Intensity*.

Interchannel Phase Deviation: The difference in timing between the left and right channel sampling times of a stereo ADC or DAC.

Interleaving: In channel coding *interleaving* is used to enhance the performance of a coder over a channel that is prone to error bursts. The basic idea behind interleaving is to spread a block of coded bits over a large number of dispersed channel symbols to allow the correction of just a few errors in each block in spite of the fact that many consecutive channel symbols are corrupted.

Interleaving is best illustrated by an example. See also *Channel Coding, Cross-Interleaved Reed-Solomon Coding.*


coded input symbol stream
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
load blocks into columns
single error correcting block
single error correcting block
single error correcting block
single error correcting block
1 6 11 16
2 7 12 17
3 8 13 18
4 9 14 19
5 10 15 20
The interleaving is accomplished by placing symbols from each block into a separate column of an array and then transmitting the symbols sequentially from the rows. For this block coding example, interleaving places symbols from separate blocks of a single error correcting code next to each other. In this way, when a burst error of 3 consecutive symbols occurs, all 3 symbols can be corrected because they come from separately coded blocks. Note that in the example below, all three symbols are from separate blocks.
read symbols from rows
burst error
interleaved symbol stream
1 6 11 16 2 7 12 17 3 8 13 18 4 9 14 19 5 10 15 20

International Electrotechnical Commission (IEC): The *IEC* was founded in 1906 with the object of promoting "international co-operation on all questions of standardization and related matters in the fields of electrical and electronic engineering and thus to promote international understanding." The *IEC* is composed of a number of committees made up from members from the main industrial countries of the world. The *IEC* publishes a wide variety of international standards and technical reports.

The *IEC* works with other international organizations, particularly with the International Organization (ISO), and also with the European Committee for Electrotechnical Standardization (CENELEC). Standards resulting from cooperations are often prefixed with the letters JTC - Joint Technical Committee. Some of the JTC standards relevant to DSP are discussed under *International Organization for Standards*.

More information on the IEC can be found at the WWW site http://133.82.181.177/ikeda/IEC/. See also *International Organization for Standards (ISO), International Telecommunication Union, Standards*.

International Mobile (Maritime) Satellite Organization (Inmarsat): *Inmarsat* provides mobile satellite communications world-wide for the maritime community. This satellite communication system supports services such as telephone, telex, facsimile, e-mail and data connections. *Inmarsat's* compact land mobile telephones (an essential tool for workers in remote parts of the world) can fit inside a briefcase and provide an excellent means of worldwide emergency communications. The various communication modes of Inmarsat rely on powerful DSP systems and the use of various coding standards.

International Organisation for Standards (ISO): *ISO* is *not* in fact an acronym for the International Organisation for Standards; that would be IOS. "ISO" is a word derived from the Greek word isos, meaning "equal" such as in words like isotropic or isosceles. However it is quite

commonplace for *ISO* to be assumed to be an acronym for International Standards Organisation, which it is not! But, on average, only one out of two authors would care.

ISO is an autonomous organization established in 1947 to promote the development of standardization worldwide. *ISO* standards essentially contain technical criteria and other detail to ensure that the specification, design, manufacture and use of materials, products, processes and services are fit for their purpose. One common example of standardization in everyday life is the woodscrew which should be produced in common *ISO* standards defining thread size, width, length etc. Another example are credit cards which should all be produced according to ISO standard widths, heights and lengths.

Standards on coding of audio and video are of particular relevance to DSP. *ISO* is made of various committees, sub-committees (SC) and working groups who oversee the definition of new standards, and ensure that current standards maintain their relevance. Some of the work most relevant to DSP is actually performed by joint technical committees (JTC) with other standards organisations such as the International Electrotechnical Commission (IEC). The ISO/IEC JTC 1 is on information technology and has the scope of standardization within established and emerging areas of information technology. Some of the key subcommittees that have been set up include:

SC 1: Vocabulary
SC 2: Coded character sets
SC 6: Telecommunications and information exchange between systems
SC 7: Software engineering
SC 11 Flexible magnetic media for digital data interchange
SC 14: Data element principles
SC 15: Volume and file structure
SC 17: Identification cards and related devices
SC 18: Document processing and related communication
SC 21: Open systems interconnection, data management and open distributed processing
SC 22: Programming languages, their environments and system software interfaces
SC 23: Optical disk cartridges for information interchange
SC 24: Computer graphics and image processing
SC 25: Interconnection of information technology equipment
SC 26: Microprocessor systems
SC 27: IT Security techniques
SC 28: Office equipment
SC 29: Coding of audio, picture, multimedia and hypermedia information
SC 30: Open electronic data interchange

Of most relevance to DSP, is the work of SC6, 24 and 29. SC29 is currently of particular interest and is responsible for standards on “Coding of Audio, Picture, Multimedia and Hypermedia Information”. SC29 is further subdivided into working groups (WG) which have already defined various standards:

WG 1: Coding of Still Pictures

ISO/IEC 11 544: JBIG (Progressive Bi-level Compression)

ISO/IEC 10 918: JPEG (Continuous-tone Still Image)

Part 1: Requirement and Guidelines
Part 2: Compliance Testing
Part 3: Extensions

WG 11: Coding of Moving Pictures and Associated Audio

ISO/IEC 11 172: MPEG-1 (Moving Picture Coding up to 1.5 Mbit/s)

Part 1: Systems
Part 2: Video
Part 3: Audio
Part 4: Compliance Testing (CD)
Part 5: Technical Report on Software for ISO/IEC 11 172

ISO/IEC 13 818: MPEG-2 (Generic Moving Picture Coding)

Part 1: Systems (CD)
Part 2: Video (CD)
Part 3: Audio (CD)
Part 4: Compliance Testing
Part 5: Technical Report on Software for ISO/IEC 13 818
Part 6: Systems Extensions
Part 7: Audio Extensions
There is also work on MPEG-4 (Very-low Bitrate Audio-Visual Coding).

WG 12: Coding of Multimedia and Hypermedia Information

ISO/IEC 13 522: MHEG (Coding of Multimedia and Hypermedia Information)

Part 1: Base Notation (ASN.1) (CD)
Part 2: Alternate Notation (SGML) (WD)
Part 3: MHEG Extensions for Scripting: Language Support

More information on the ISO and ISO JTC standards can be found in the relevant ISO publications which are summarized on http://www.iso.ch. See also *International Electrotechnical Commission (IEC), International Telecommunication Union (ITU), Standards*.

International Standards Organization: See *International Organisation for Standards.*

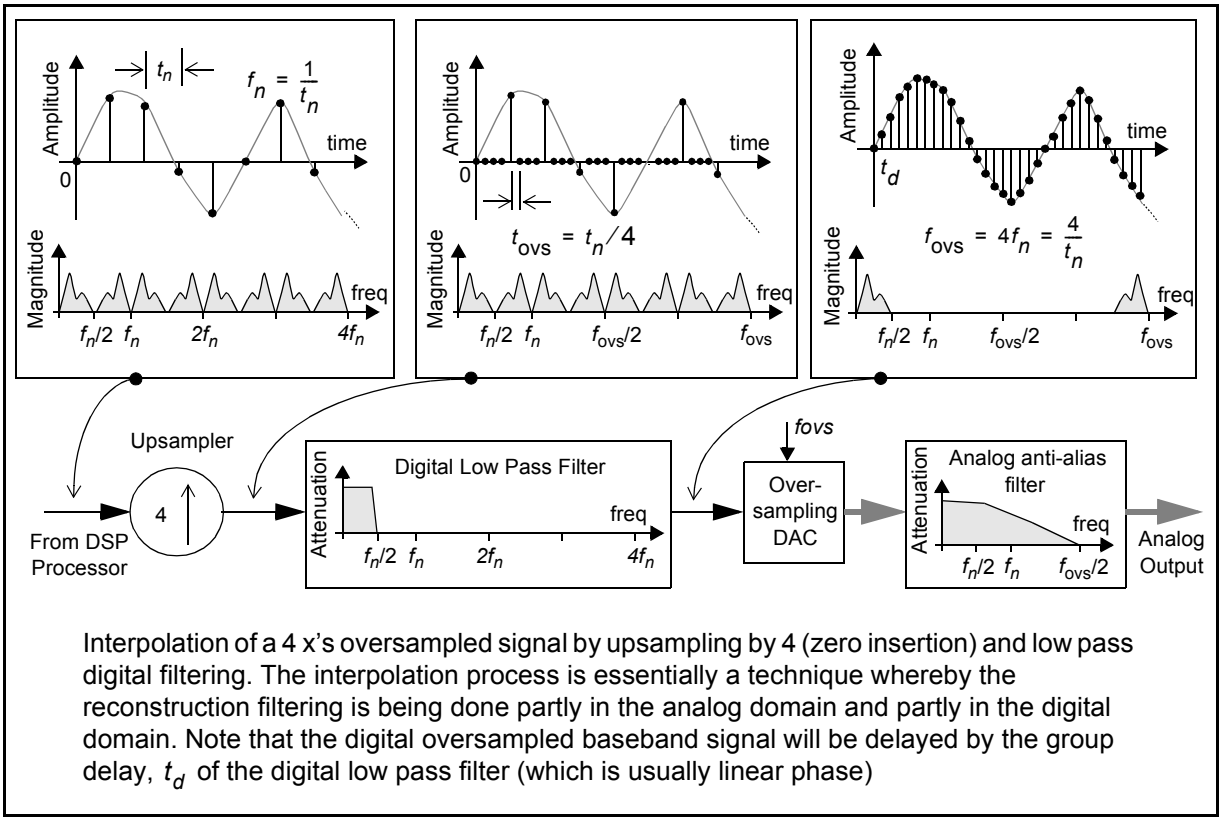
International Telecommunication Union (ITU): The ITU is an agency of the United Nations who operate a world-wide organization from which governments and private industry from various countries coordinate the definition, implementation and operation of telecommunication networks and services. The responsibilities of the ITU extend to regulation, standardization, coordination and development of international telecommunications. They also have a general responsibility to ensure the integration of the differing policies and systems in various countries. The headquarters of the ITU is currently *International Telecommunication Union*, Place des Nations, CH-1211 Geneva 20, Switzerland. They can be contacted on the world wide web at address http://www.itu.ch.

The recommendations and various standards of the ITU are divided into two key areas resulting from the output two advisory committees on: (1) Telecommunication and denoted as ITU-T recommendations, (formerly known as CCITT); and (2) Radiocommunications and denoted as ITU-R recommendations (formerly known as CCIR. See also *International Organisation for Standards, ITU-R Recommendations, ITU-T Recommendations, Multimedia Standards, Standards*.

Internet: The name give to the worldwide connection of computers each having a unique identifying *internet* number. The internet currently allows interchange of electronic mail, and general computer files containing anything from text, images, and audio. Useful tools for navigating the internet and exploring information available from other users on machines other than your own, include ftp (file transfer protocol) Gopher, Netscape, Mosaic, and Lynx [169], etc.

Interpolation: Interpolation is the creation of intermediate discrete values between two samples of a signal. For example, if 3 intermediate and equally spaced samples are created, then the sampling

rate has increased by a factor of 4. *Interpolation* is usually accomplished by first up-sampling to insert zeroes between existing samples, and then filtering with a low pass digital filter.



Interpolation of a 4 x's oversampled signal by upsampling by 4 (zero insertion) and low pass digital filtering. The interpolation process is essentially a technique whereby the reconstruction filtering is being done partly in the analog domain and partly in the digital domain. Note that the digital oversampled baseband signal will be delayed by the group delay, t_d of the digital low pass filter (which is usually linear phase)

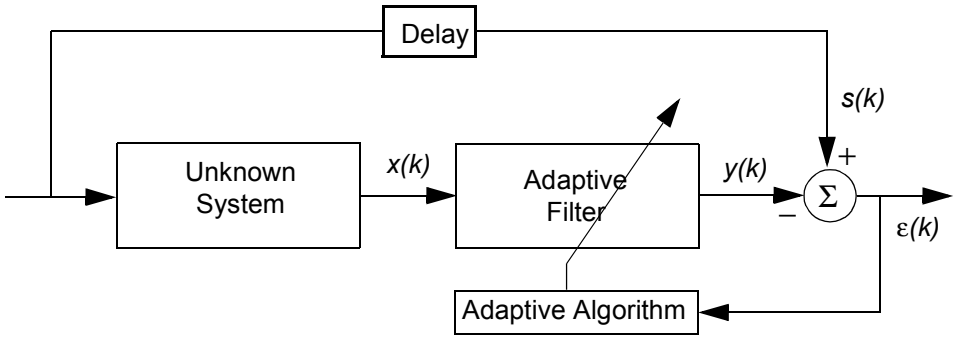
Other types of curve fitting interpolators can also be produced, although there are less common. Interpolators are widely found in digital audio systems such as CD players, where oversampling filters (typically 4 ×'s) are used to increase the sampling rate in order to allow a simpler reconstruction filter to be used at the output of the digital to analog converter (DAC). See also *Upsampling, Decimation, Downsampling, First Order Hold*, *Fractional Sampling Rate Conversion, Zero Order Hold.*

Interrupt: Inside a DSP processor an *interrupt* will temporarily halt the processor and force it to perform an interrupt routine. For example an *interrupt* may happen every $1/f_s$ seconds in order that a DSP processor executes the *interrupt* service routine, whereby it reads the value from an A/D converter at a rate of f_s samples every second.

Inverse, Matrix: See *Matrix Operations - Inverse.*

Inverse System Identification: Using adaptive filtering techniques, the approximate inverse of an unknown filter, plant or data channel can be identified. In an adaptive signal processing *inverse system identification architecture*, when the error, $\varepsilon(k)$ has adapted to a minimum value (ideally zero) then this means that in some sense $y(k) \approx s(k)$, where *s(k)* is the input to the unknown channel. Therefore the transfer function of the adaptive filter is now an approximate inverse of the unknown system. *Inverse system identification* is widely used for equalizing data transmission

channels. See also *Adaptive Filtering, Adaptive Line Enhancer, Equalisation, Least Mean Squares Algorithm, System Identification,*



Generic Adaptive Signal Processing Inverse System Identification Architecture

Inversion Lemma: See *Matrix Properties - Inversion Lemma.*

ITU-R Recommendations: The International Telecommunication Union (ITU) have produced a very comprehensive set of regulatory, standardizing and coordination documents for radiocommunication systems. The ITU-Radiocommunications (ITU-R) advisory committee are responsible for the generation, upkeep and amendment of the *ITU-R recommendations.* These recommendations are classified into various subgroups or series identified by the letters:

Series	*Description*
BO	*Broadcasting satellite service (sound and television);*
BR	*Sound and television recording;*
BS	*Broadcasting service (sound);*
BT	*Broadcasting service (television);*
F	*Fixed service;*
IS	*Inter-service sharing and compatibility;*
M	*Mobile, radiodetermination, amateur and related satellite services;*
Pl	*Propagation in ionized media;*
PN	*Propagation in non-ionized media;*
RA	*Radioastronomy;*
S	*Fixed satellite service;*
SA	*Space applications;*
SF	*Frequency sharing between the fixed satellite service and the fixed service;*
SM	*Spectrum management techniques;*
SNG	*Satellite news gathering;*
TF	*Time signals and frequency standards emissions;*
V	*Vocabulary and related subjects.*

In addition to the ITU-R (radiocommunication) recommendations, there are also the ITU-T (telecommunication) recommendations See also *International Organization for Standards*, *International Telecommunication Union, ITU-T Recommendations, Standards*.

ITU-T Recommendations: The International Telecommunication Union (ITU) have produced a very comprehensive set of regulatory, standardizing and coordination documents for telecommunication systems. The ITU-Telecommunications (ITU-T) advisory committee are responsible for the generation, upkeep and amendment of the *ITU-T recommendations.* These

standards, definitions and recommendations are classified into various subgroups or series identified by a letter:

A	*Organization of the work of the ITU-T;*
B	*Means of expression (definitions, symbols, classification);*
C	*General telecommunication statistics;*
D	*General tariff principles;*
E	*Overall network operation (numbering, routing, network management, etc.;*
F	*Services other than telephone (ops, quality, service definitions and human factors);*
G	*Transmission systems and media, digital systems and networks;*
H	*Line transmission of non-telephone signals;*
I	*Integrated Services Digital Networks;*
J	*Transmission of sound programmes and television signals;*
K	*Protection against interference;*
L	*Construction, installation and protection of cable and other elements of outside plant;*
M	*Maintenance: international systems, telephone, telegraphy, fax & leased circuits;*
N	*Maintenance: international sound programme and television transmission circuits;*
O	*Specifications of measuring equipment;*
P	*Telephone transmission quality, telephone installations, local line networks;*
Q	*Switching and Signalling;*
R	*Telegraph transmission;*
S	*Telegraph services terminal equipment;*
T	*Terminal characteristics protocols for telematic services, document architecture;*
U	*Telegraph switching;*
V	*Data communication over the telephone network;*
X	*Data networks and open system communication;*
Z	*Programming languages.*

These recommendations were formerly known as CCITT (the former name of the ITU) regulations, and are available from the ITU (usually for a price) in published book form (20 volumes and 61 Fascicles), or electronic form (http://www.itu.ch). The book form is also sometimes referred to as the “blue book”.

The work of the committee is clearly outlined in the A-series recommendations:

A.1	Presentation of contributions relative to the study of questions assigned to the ITU-T.
A.10	Terms and definitions.
A.12	Collaboration with the International Electrotechnical Commission (IEC) on the subject of definitions for telecommunications.
A.13	Collaboration with the IEC on graphical symbols and diagrams used in telecommunications.
A.14	Production maintenance and publication of ITU-T terminology.
A.15	Elaboration and presentation of texts for Recommendations of the ITU Telecommunication Standardization Sector.
A.20	Collaboration with other international organizations over data transmission.
A.21	Collaboration with other international organizations on ITU-T defined telematic services.
A.22	Collaboration with other international organizations on information technology.
A.23	Collaboration with other international organizations on information technology, telematic services and data transmission.
A.30	Major degradation or disruption of service.

From a DSP algorithm and implementation perspective the G-series specifies a variety of algorithms for audio digital signal coding and compression, the H-series specifies video compression techniques and the V-series specifies modem data communications strategies including echo cancellation, equalisation and data compression.

In addition to the ITU-T (*tele*communication) recommendations, there are also the ITU-T (*radio*communication) recommendations. See also *G-Series Recommendations, H-Series*

Recommendations, International Organization for Standards, International Telecommunication Union, ITU-R Recommendations, MPEG, Standards, V-Series Recommendations.

i860: Intel's powerful RISC processor which has been used in many DSP applications.

J

j: The electrical engineering representation of $\sqrt{-1}$, the imaginary number that mathematicians denote as *"i"*. However, electrical engineers use *"i"* to denote current.

JND: Just Noticeable Difference. See *Difference Limen.*

J-Series Recommendations: The J-series telecommunication recommendations from the International Telecommunication (ITU), advisory committee on telecommunications (denoted ITU-T and formerly known as CCITT) provide standards for transmission of sound programme and television signals. Some of the current recommendations (http://www.itu.ch) include:

- J.11 Hypothetical reference circuits for sound-programme transmissions.
- J.12 Types of sound-programme circuits established over the international telephone network.
- J.13 Definitions for international sound-programme circuits.
- J.14 Relative levels and impedances on an international sound-programme connection.
- J.15 Lining-up and monitoring an international sound-programme connection.
- J.16 Measurement of weighted noise in sound-programme circuits.
- J.17 Pre-emphasis used on sound-programme circuit.
- J.18 Crosstalk in sound-programme circuits set up on carrier systems.
- J.19 A conventional test signal simulating sound-programme signals for measuring interference in other channels.
- J.21 Performance characteristics of 15 kHz-type sound-programme circuits - circuits for high quality monophonic and stereophonic transmissions.
- J.23 Performance characteristics of 7 kHz type (narrow bandwidth) sound-programme circuits.
- J.31 Characteristics of equipment and lines used for setting up 15 kHz type sound-programme circuits.
- J.33 Characteristics of equipment and lines used for setting up 6.4 kHz type sound-programme circuits.
- J.34 Characteristics of equipment used for setting up 7 kHz type sound-programme circuits
- J.41 Characteristics of equipment for the coding of analogue high quality sound programme signals for transmission on 384 kbit/s channels.
- J.42 Characteristics of equipment for the coding of analogue medium quality sound-programme signals for transmission on 384-kbit/s channels.
- J.43 Characteristics of equipment for the coding of analogue high quality sound programme signals for transmission on 320 kbit/s channels.
- J.44 Characteristics of equipment for the coding of analogue medium quality sound-programme signals for transmission on 320 kbit/s channels.
- J.51 General principles and user requirements for the digital transmission of high quality sound programmes.
- J.52 Digital transmission of high-quality sound-programme signals using one, two, or three 64 kbit/s channels per mono signal (and up to six per stereo signal).
- J.61 Transmission performance of television circuits designed for use in international connections.
- J.62 Single value of the signal-to-noise ratio for all television systems.
- J.63 Insertion of test signals in the field-blanking interval of monochrome and colour television signals.
- J.64 Definitions of parameters for simplified automatic measurement of television insertion test signals.
- J.65 Standard test signal for conventional loading of a television channel.
- J.66 Transmission of one sound programme associated with analogue television signal by means of time division multiplex in the line synchronizing pulse.
- J.67 Test signals and measurement techniques for transmission circuits carrying MAC/packet signals for HD-MAC signals.
- J.73 Use of a 12-MHz system for the simultaneous transmission of telephony and television.
- J.74 Methods for measuring the transmission characteristics of translating equipments.
- J.75 Interconnection of systems for television transmission on coaxial pairs and on radio-relay links.
- J.77 Characteristics of the television signals transmitted over 18 MHz and 60-MHz systems.

J.80 Transmission of component-coded digital television signals for contribution-quality applications at bit rates near 140 Mbit/s.
J.81 Transmission of component-coded television signals for contribution-quality applications at the third hierarchical level of ITU-T Recommendation G.702.
J.91 Technical methods for ensuring privacy in long-distance international television transmission.

For additional detail consult the appropriate standard document or contact the ITU. See also *International Telecommunication Union, ITU-T Recommendations, Standards*.

Joint Bi-level Image Group (JBIG): *JBIG* is the name for a lossless compression algorithm for binary (one bit/pixel) images which results from the International Organization for Standards (ISO) sub-committee (SC) 29 which is responsible for standards on "Coding of Audio, Picture, Multimedia and Hypermedia Information". Working Group (WG) 1 of SC29 (ISO/IEC JTC1/SC29/WG1) considered the problem of coding of still binary images and produced a joint standard with the International Electrotechnical Commission (IEC): ISO/IEC 10918 - JBIG (Progressive Bi-level Compression).

JBIG is intended to replace the current, (and less effective) Group 3 and 4 fax algorithms which are primarily used for document text transmission (i.e., Fax). JBIG achieves compression by modelling the redundancy in the image as the correlations of the pixel currently being coded with a set of nearby pixels using arithmetic coding techniques. See also *JPEG, MPEG Standards, Standards*.

Joint Photographic Experts Group (JPEG): *JPEG* is the general name for a lossy compression algorithm for continuous tone still images. *JPEG* is the original name of the committee who drafted the standard for the International Organization for Standards (ISO) sub-committee (SC) 29 which is responsible for standards on "Coding of Audio, Picture, Multimedia and Hypermedia Information". Working Group (WG) 1 (ISO/IEC JTC1/SC24/WG1) considered the problem of coding of still binary images and produced the JPEG joint standard with the International Electrotechnical Commission (IEC): ISO/IEC 11544 - *JPEG* (Continuous Tone Still Image).

JPEG is designed for compressing full 24 bit colour or gray-scale digital images of "natural" (real-world) scenes (as opposed to, for example, complex geometrical patterns). *JPEG* does not cater for motion picture compression (see MPEG) or for black and white image compression (see JBIG) where is does not cope well with edges formed at black-white boundaries. The primary compression scheme in JPEG consists of a two dimensional discrete cosine transform (DCT) of image blocks, a coefficient quantizer, a zig-zag scan of the quantized DCT coefficients (that has probably produced long runs of zeros) that is subsequently run-length encoded by a Huffman code designed for a set of training image zig-zag scan fields [39]. *JPEG* is a lossy algorithm however most of the compression is achieved by exploiting known limitations of the human eye, for example that small colour details are not perceived by the eye and brain as well as small details of light and dark.

The degree of information loss from *JPEG* compression can be varied by adjusting the values of certain compression parameters. Therefore file size can be traded off against image quality, which will of course depend on the actual application. Extremely small files (thumbnails) can be produced using *JPEG* which are useful for icons or image indexing and archive purposes.

The ITU-T T-series standards T.80 - T83 are similar to *JPEG*:

- T.80 Common components for image compression and communication; basic principles.

- T.81 Digital compression and encoding of continuous tone still images.
- T.82 Progressive compression techniques for bi-level images.
- T.83 Compliance testing.

Additional information is available form the independent *JPEG* group at `jpeg-info@uunet.uu.net`. JPEG software and file specifications are available from a number of FTP sites, including ftp://ftp.uu.net:/graphics/jpeg. See also *JBIG, MPEG, Standards, T-Series Recommendations*.

Joint Stereo Coding: When compressing hifidelity stereo audio higher levels of compression can be obtained by exploiting the commonalities between the audio on the left and right channels, than would be gained by compressing the left and right channels independently. MPEG-Audio has a *joint stereo coding* facility. See *Compression, Moving Picture Experts Group (MPEG) - Audio*.

Just (Music) Scale: A few hundred years ago, prior to the existence of the equitemporal or Western music scale, a (major) musical key was formed from using certain carefully chosen frequency ratios between adjacent notes, rather than the constant tone and semitone ratios of the modern Western music scale. The C-major *just scale* would have had the following frequency ratios:

C-major Scale	**C**	**D**	**E**	**F**	**G**	**A**	**B**	**C**
Frequency ratio	**1/1**	**9/8**	**5/4**	**4/3**	**3/2**	**5/3**	**15/8**	**2/1**

The frequency ratio gives the ratio of the fundamental frequency of the root note, to the current note. The above ratios correspond to the Just Music Scale.

Any note can be used to realise a just major key or scale. However using the *just scale* it is difficult to form other major or minor keys without a complete retuning of the instrument as all of the fundamental frequencies in other keys are different. Instruments that are tuned and played using the *just scale* will probably sound in some sense “medieval” as our modern appreciation of music is now firmly based on the equitempered Western music scale. See also *Digital Audio, Music, Music Synthesis, Pythagorean Scale, Western Music Scale.*

Just Noticeable Difference: See *Difference Limen.*

K

k: ”*k*” (along with “i” and “n”) is often used as a discrete time index for in DSP notation. It is also often used as the frequency index in the DFT. See *Discrete Time, Discrete Fourier Transform*.

Karaoke DSP: For professionally recorded stereo music on CDs, DATs and so on, the vocal track, $v(k)$, of a song is usually centered on the left and right channels, i.e. the same signal in the left track $L(k)$ and the right track $R(k)$ which is perceived as coming from between the two loudspeakers if the listener is sitting equidistant from both. The musical instruments are likely to be laid out in some off-centre set up which means that they are unlikely to be identical signals on both left and right channels, i.e.:

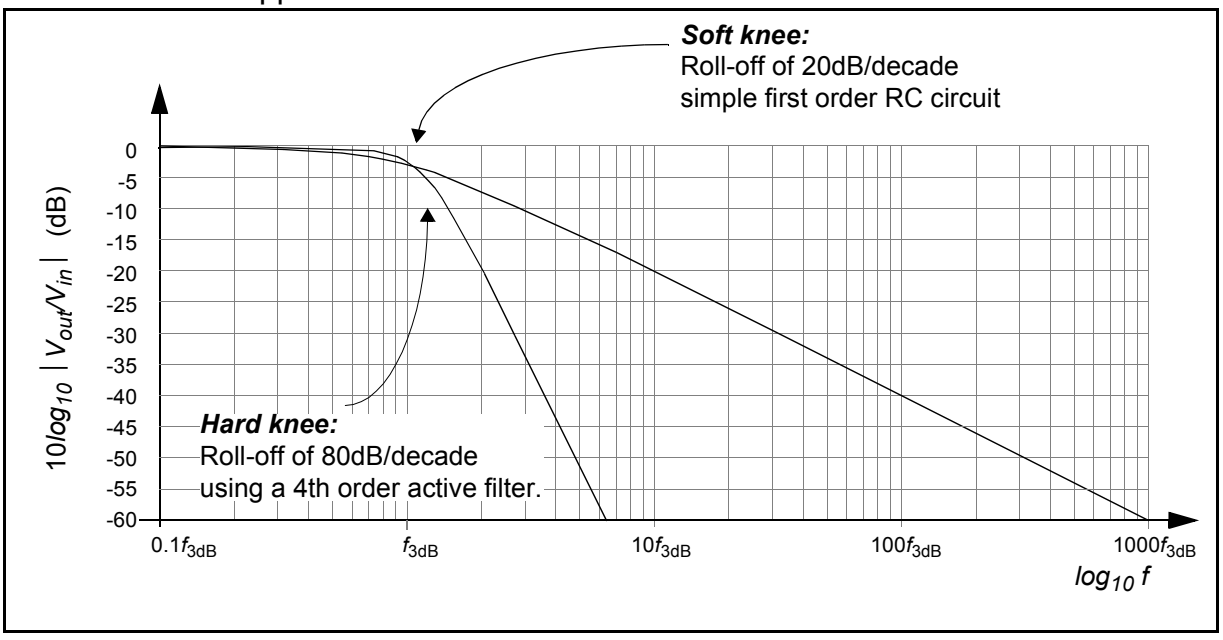
$$\text{Left} = L(k) = v(k) + M_L(k) \qquad \text{Right} = R(k) = v(k) + M_R(k) \tag{222}$$

By digitally subtracting the left and right channels:

$$L(k) - R(k) = M_L(k) - M_R(k) \tag{223}$$

the vocal track may be somewhat attenuated, enabling the song to be played with the vocals de-emphasised by a few dBs, all ready for the bellowing tones of a *Karaoke* singer! See *My Way by Frank Sinatra*.

Knee: The *knee* is the part of a magnitude-frequency graph of a filter, where the transition from passband to stopband is made. A soft *knee* is where the transition realises a filter with very low roll-off, and a harder *knee* approaches the ideal filter. See also *Roll-off* .



Khoros: *Khoros* is a block diagram simulator for image and video processing which runs on a variety of computer platforms such as Sun workstations.

Kronecker Impulse, or Kronecker Delta Function: See *Unit Impulse Function*.

Kronecker Product: See *Matrix Operations - Kronecker Product.*

L

LA (Linear Arithmetic) Synthesis: A technique for synthesis of the sound of musical instruments [32]. See also *Music, Music Synthesis*.

LabView: A software package from National Instruments Inc. which allows powerful PC based DSP instrumentation front-ends to be designed. *LabView* also convincingly presents the Virtual Instrument concept. See also *Virtual Instrument*.

Laplace: A mathematical transform use for the analysis of analog systems.

Laplacian: A probability distribution that is often used to model the differences between adjacent pixels in an image.

Lateralization: *Lateralization* refers to a psychoacoustics task in which a sound is determined to be at some point *within* the head, either near one ear or the other along a line separating the two ears. Very much like localization, *lateralization* differs in that the sound source is perceived within the head rather than outside of the head. The common experience of listening to stereophonic music via headphones (*lateralization*) versus listening to the same music via loud speakers in a normal room (localization) emphasizes the difference between the two tasks. See also *Localization.*

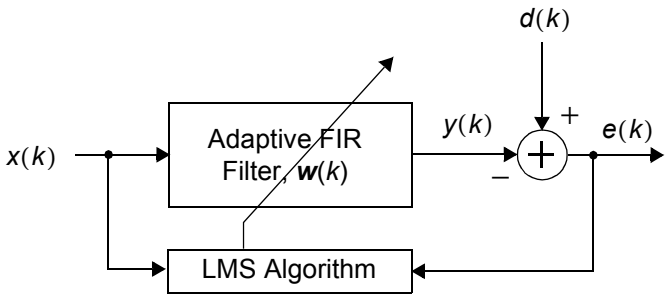
Law of First Wavefront: In a reverberant environment the sound energy received by the direct path can be very much lower than the energy received by indirect reflective paths. However the human ear is still able to localize the sound location correctly by localizing the first components of the signal to arrive. Later echoes arriving at the ear increase the perceived loudness of the sound as they will have the same general spectrum. This psychoacoustic effect is sometimes known as the *law of the first wavefront* or the Haas effect, and more commonly the precedence effect. The precedence effect applies mainly to short duration sounds or those of a discontinuous or varying form. See also *Ear, Lateralization, Source Localization, Threshold of Hearing.*

LDU: See *Matrix Decompositions - LDU Decomposition*.

Leaky LMS: See *Least Mean Squares Algorithm Variants.*

Least Mean Squares (LMS) Algorithm: The *LMS* is an adaptive signal processing algorithm that is very widely used in adaptive signal processing applications such as system identification, inverse system identification, noise cancellation and prediction. The *LMS algorithm* is very simple to implement in real time and in the mean will adapt to a neighborhood of the Wiener-Hopf least mean square solution. The *LMS algorithm* can be summarised as follows:

To derive the *LMS algorithm*, first consider plotting the mean squared error (MSE) performance surface (i.e. $E\{e^2(k)\}$ as a function of the weight values) which gives an N+1-dimensional hyperparaboloid which has one minimum. It is assumed that $x(k)$ (the input data sequence)and $d(k)$ (a desired signal) are wide sense stationary signals (see *Wiener-Hopf Equations*). For



- $y(k) = \sum_{n=0}^{N-1} w(k)x(k-n) = \boldsymbol{w}^T(k)\boldsymbol{x}(k)$

  where $\boldsymbol{x}(k) = [x(k), x(k-1), x(k-2), \ldots, x(k-N+2), x(k-N+1)]^T$

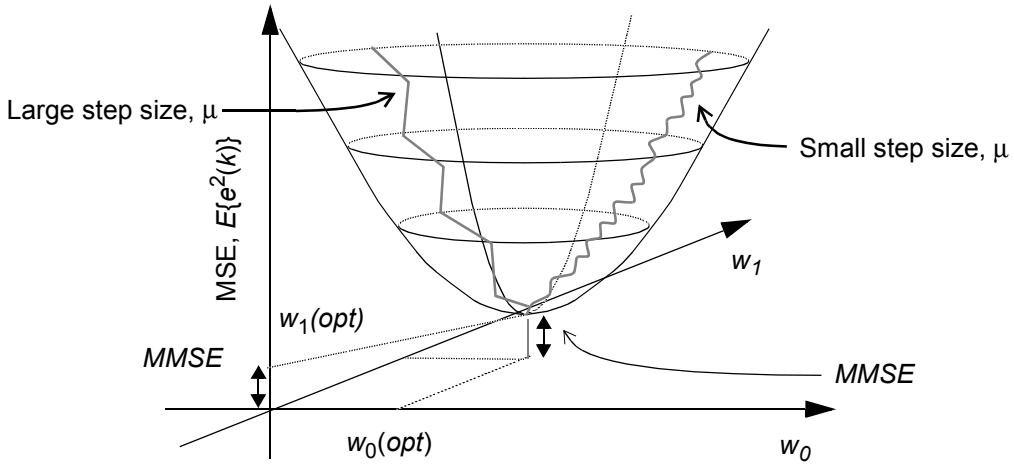
  $\boldsymbol{w}(k) = [w_0(k), w_1(k), w_2(k), \ldots, w_{N-2}(k), w_{N-1}(k)]^T$

- $e(k) = d(k) - y(k) = d(k) - \boldsymbol{w}^T(k)\boldsymbol{x}(k)$

- $\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu e(k)\boldsymbol{x}(k)$

In the generic adaptive filtering architecture the aim can intuitively be described as adapting the impulse response of the FIR digital filter such that the input signal $x(k)$ is filtered to produce $y(k)$ which, when subtracted from desired signal $d(k)$, will minimise the error signal $e(k)$. If the filter weights are updated using the LMS weight update then the adaptive FIR filter will adapt to the minimum mean squared error, assuming $d(k)$ and $x(k)$ to be wide sense stationary signals.

discussion and illustration purposes the three dimensional paraboloid for a two weight FIR filter can be drawn:



The mean square error (MSE) performance surface for a two weight FIR filter. The Wiener-Hopf solution is denoted as $w_k(opt))$, which denotes where the minimum MSE (MMSE) occurs. The gradient based LMS algorithm will (on average) adapt towards the MMSE by taking "jumps" in the direction of the negative of the gradient of the surface (therefore "downhill").

To find the minimum mean squared error (MMSE) we can use the Wiener Hopf equation, however this is an expensive solution in computation terms. As an alternative we can use gradient based techniques, whereby we can traverse down the inside of the parabola by using an iterative algorithm

which always updates the filter weights in the direction opposite of the steepest gradient. The iterative algorithm is often termed gradient descent and has the form:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu(-\underline{\nabla}_k) \tag{224}$$

where $\underline{\nabla}_k$ is the gradient of the performance surface:

$$\begin{aligned} \underline{\nabla}_k &= \frac{\partial}{\partial \boldsymbol{w}(k)} E\{e^2(k)\} \\ &= 2\boldsymbol{R}\boldsymbol{w}(k) - 2\boldsymbol{p} \end{aligned} \tag{225}$$

where $\boldsymbol{R}$ is the correlation matrix, $\boldsymbol{p}$ is cross correlation vector (see *Correlation Matrix* and *Cross Correlation Vector*) and μ is the step size (used to control the speed of adaption and the achievable minimum or misadjustment). In the above figure a small step size "jumps" in small steps towards the minimum are is therefore slow to adapt, however the small jumps mean that it will arrive very close to the MMSE and continue to jump back and forth close to the minimum. For a large step size the jumps are larger and adaption to the MMSE is faster, however when the weight vector reaches the bottom of the bowl it will jump back and forth around the MMSE with a large magnitude than for the small step size. The error caused by the traversing of the bottom of the bowl is usually called the excess mean squared error (EMSE).

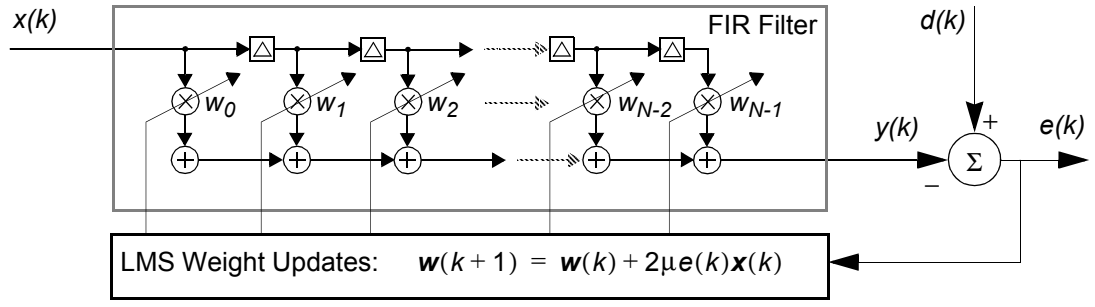
To calculate the MSE performance surface gradient directly is (like the Wiener Hopf equation) very expensive as it requires that both $\boldsymbol{R}$, the correlation matrix and $\boldsymbol{p}$, the cross correlation vector are known (see *Wiener-Hopf Equations*). In addition, if we knew $\boldsymbol{R}$ and $\boldsymbol{p}$, we could directly compute the optimum weight vector. But in general, we do not have access to $\boldsymbol{R}$ and $\boldsymbol{p}$. Therefore a subtle innovation, first defined for DSP by Widrow et al [152], was to replace the actual gradient with an instantaneous (noisy) gradient estimate. One approach to generating this noisy gradient estimate is to take the gradient of the actual squared error (versus the mean squared error), i.e.

$$\begin{aligned} \hat{\underline{\nabla}}_k &= \frac{\partial}{\partial \boldsymbol{w}(k)} e^2(k) \\ &= 2e(k)\frac{\partial}{\partial \boldsymbol{w}(k)} e(k) = -2e(k)\frac{\partial}{\partial \boldsymbol{w}(k)} y(k) = -2e(k)\boldsymbol{x}(k) \end{aligned} \tag{226}$$

Therefore using this estimated gradient, $\hat{\underline{\nabla}}_k$ in the gradient descent equation yields the *LMS algorithm*:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu e(k)\boldsymbol{x}(k) \tag{227}$$

The *LMS* is very straightforward to implement and only requires *N* multiply-accumulates (MACs) to perform the FIR filtering, and *N* MACs to implement the *LMS* equation. A typical signal flow graph for the *LMS* is shown below:



A simple signal flow graph for an adaptive FIR filter, where the adaptive nature of the filter weights is explicitly illustrated.

The *LMS* is very widely used in many applications such as telecommunications, noise control, control systems, biomedical DSP, and so on. Its properties have been very widely studied and a good overview can be found in [77], [53].

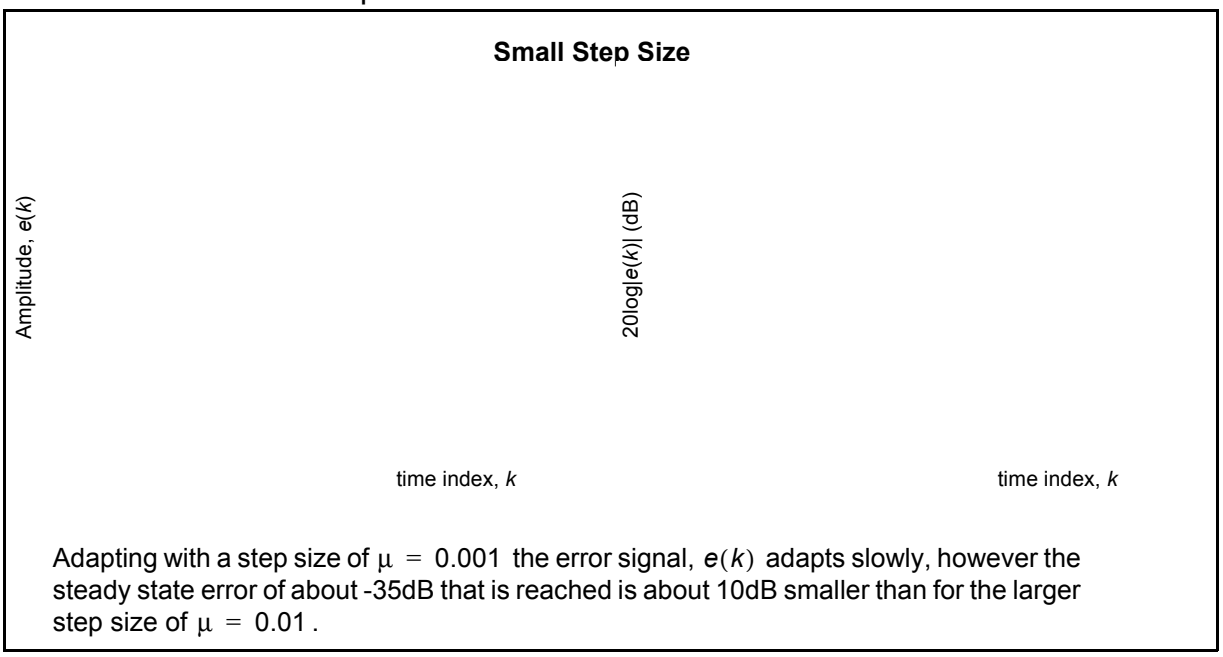
From a practical implementation point of view the algorithm designer must carefully choose the filter length to suit the application. In addition, the step size must be chosen to ensure stability and a good convergence rate. For the *LMS* upper and lower bounds for the adaptive step size can be calculated as:

$$0 < \mu < \frac{1}{NE\{x^2(k)\}} \cong \qquad 0 < \mu < \frac{1}{N\{\text{Input Signal Power}\}} \tag{228}$$

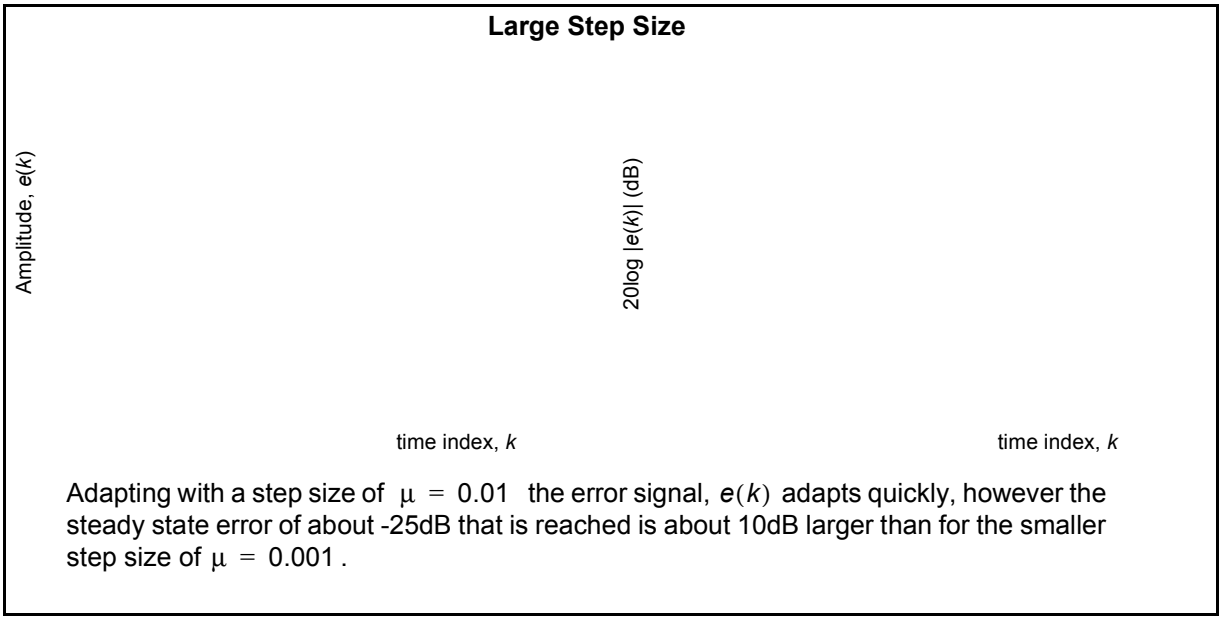
A more formal bound can be defined in terms of the eigenvalues of the input signal correlation matrix [53]. However for practical purposes these values are not calculated and the above practical bound is used (see *Least Mean Squares Algorithm Convergence*).

In general the speed of adaption is inversely proportional to the step size, and the excess MSE or steady state error is proportional to the step size. A simple example of a 20 weight FIR filter being

used to identify an unknown filter (i.e., system identification) was simulated to produce the error plots below for two different step sizes of 0.001 and 0.01:



Adapting with a step size of $\mu = 0.001$ the error signal, $e(k)$ adapts slowly, however the steady state error of about -35dB that is reached is about 10dB smaller than for the larger step size of $\mu = 0.01$.



Adapting with a step size of $\mu = 0.01$ the error signal, $e(k)$ adapts quickly, however the steady state error of about -25dB that is reached is about 10dB larger than for the smaller step size of $\mu = 0.001$.

Clearly a trade-off exists -- once again the responsibility of choosing this parameter is in the domain of the algorithm designer. See also *Acoustic Echo Cancellation, Active Noise Control, Adaptive Line Enhancer, Adaptive Signal Processing, Adaptive Step Size, Correlation Matrix, Correlation Vector, Echo Cancellation, Least Mean Squares Algorithm Convergence, Least Mean Squares Algorithm Misadjustment/Algorithm/IIR Algorithms/Time Constant/ Variants, Least Mean Squares Filtered-X Algorithm, Least Squares, Noise Cancellation, Recursive Least Squares, Wiener-Hopf Equations, Volterra Filter.*

Least Mean Squares (LMS) Algorithm Convergence: It can be shown that the (noisy) gradient estimate used in the LMS algorithm (see *Least Mean Squares Algorithm*) is an unbiased estimate of the true gradient:

$$
\begin{aligned}
E\left\{\hat{\underline{\nabla}}_k\right\} &= E[-2e(k)\boldsymbol{x}(k)] \\
&= (E[-2(d(k)-\boldsymbol{w}^T(k)\boldsymbol{x}(k))\boldsymbol{x}(k)]) \\
&= 2\boldsymbol{R}\boldsymbol{w}(k)-2\boldsymbol{p} \\
&= \hat{\underline{\nabla}}_k
\end{aligned}
\tag{229}
$$

where we have assumed that $\boldsymbol{w}(k)$ and $\boldsymbol{x}(k)$ are statistically independent.

It can be shown that in the mean the LMS will converge to the Wiener-Hopf solution if the step size, μ, is limited by the inverse of the largest eigenvalue. Taking the expectation of both sides of the LMS equation gives:

$$
\begin{aligned}
E\{\boldsymbol{w}(k+1)\} &= E\{\boldsymbol{w}(k)\}+2\mu E[e(k)\boldsymbol{x}(k)] \\
&= E\{\boldsymbol{w}(k)\}+2\mu(E[d(k)\boldsymbol{x}(k)]-E[(\boldsymbol{x}(k)\boldsymbol{x}^T(k))\boldsymbol{w}(k)])
\end{aligned}
\tag{230}
$$

and again assuming that $\boldsymbol{w}(k)$ and $\boldsymbol{x}(k)$ are statistically independent:

$$
\begin{aligned}
E\{\boldsymbol{w}(k+1)\} &= E\{\boldsymbol{w}(k)\}+2\mu(\boldsymbol{p}-\boldsymbol{R}E\{\boldsymbol{w}(k)\}) \\
&= (\boldsymbol{I}-2\mu\boldsymbol{R})E\{\boldsymbol{w}(k)\}+2\mu\boldsymbol{R}\boldsymbol{w}_{opt}
\end{aligned}
\tag{231}
$$

where $\boldsymbol{w}_{opt} = \boldsymbol{R}^{-1}\boldsymbol{p}$ and $\boldsymbol{I}$ is the identity matrix. Now, defining $\boldsymbol{v}(k) = \boldsymbol{w}(k)-\boldsymbol{w}_{opt}$ then we can rewrite the above in the form:

$$
E\{\boldsymbol{v}(k+1)\} = (\boldsymbol{I}-2\mu\boldsymbol{R})E\{\boldsymbol{v}(k)\} \tag{232}
$$

For convergence of the LMS to the Wiener-Hopf, we require that $\boldsymbol{w}(k)\rightarrow\boldsymbol{w}_{opt}$ as $k\rightarrow\infty$, and therefore $\boldsymbol{v}(k)\rightarrow\boldsymbol{0}$ as $k\rightarrow\infty$. If the eigenvalue decomposition of $\boldsymbol{R}$ is given by $\boldsymbol{Q}^T\Lambda\boldsymbol{Q}$, where $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$ and Λ is a diagonal matrix then writing the vector $\boldsymbol{v}(k)$ in terms of the linear transformation $\boldsymbol{Q}$, such that $E\{\boldsymbol{v}(k)\} = \boldsymbol{Q}^TE\{\boldsymbol{u}(k)\}$ and multiplying both sides of the above equation, we realise the decoupled equations:

$$
E\{\boldsymbol{u}(k+1)\} = (\boldsymbol{I}-2\mu\Lambda)E\{\boldsymbol{u}(k)\} \tag{233}
$$

and therefore:

$$
E\{\boldsymbol{u}(k+1)\} = (\boldsymbol{I}-2\mu\Lambda)^kE\{\boldsymbol{u}(0)\} \tag{234}
$$

where $(\boldsymbol{I}-2\mu\Lambda)$ is a diagonal matrix:

$$(\boldsymbol{I} - 2\mu\Lambda) = \begin{bmatrix} (1-2\mu\lambda_0) & 0 & 0 & \dots & 0 \\ 0 & (1-2\mu\lambda_1) & 0 & \dots & 0 \\ 0 & 0 & (1-2\mu\lambda_2) & \dots & 0 \\ : & : & : & \dots & 0 \\ 0 & 0 & 0 & 0 & (1-2\mu\lambda_{N-1}) \end{bmatrix} \tag{235}$$

For convergence of this equation to the zero vector, we require that

$$(1-2\mu\lambda_n)^n \to 0 \quad \text{for all} \quad n = 0, 1, 2, \dots, N-1 \tag{236}$$

Therefore the step size, μ, must cater for the largest eigenvalue, $\lambda_{max} = \max(\lambda_0, \lambda_1, \lambda_2, \dots, \lambda_{N-1})$ such that: $|1 - 2\mu\lambda_{max}| < 1$, and therefore:

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{237}$$

This bound is a necessary and sufficient condition for convergence of the algorithm in the mean square sense. However, this bound is not convenient to calculate, and hence not particularly useful for practical purposes. A more useful sufficient condition for bounding μ can be found using the linear algebraic result that:

$$\text{trace}[\boldsymbol{R}] = \sum_{n=0}^{N-1} \lambda_n \tag{238}$$

i.e. the sum of the diagonal elements of the correlation matrix $\boldsymbol{R}$, is equal to the sum of the eigenvalues, then the inequality:

$$\lambda_{max} \leq \text{trace}[\boldsymbol{R}] \tag{239}$$

will hold. However if the signal $x(k)$ is wide sense stationary, then the diagonal elements of the correlation matrix, $\boldsymbol{R}$, are $E\{x^2(k)\}$ which is a measure of the signal power. Hence:

$$\text{trace}[\boldsymbol{R}] = NE\{x^2(k)\} = N\text{<Signal Power>} \tag{240}$$

and the well known LMS stability bound (sufficient condition) of:

$$0 < \mu < \frac{1}{NE[x_k^2]} \tag{241}$$

is the practical result. See also *Adaptive Signal Processing, Least Mean Squares Algorithm, Least Mean Squares Algorithm Misadjustment, Least Mean Squares Algorithm Time Constant, Wiener-Hopf Equations.*

Least Mean Squares (LMS) Algorithm Misadjustment: *Misadjustment* is a term used in adaptive signal processing to indicate how close the achieved mean squared error (MSE) is to the

minimum mean square error. It is defined as the ratio of the excess MSE, to the minimum MSE, and therefore gives a measure of how well the filter can adapt. For the LMS:

$$\begin{aligned}\text{Misadjustment} &= \frac{\text{excess MSE}}{\text{MMSE}} \\ &\approx \mu \text{trace}[\boldsymbol{R}] \\ &\approx \mu N \text{<Signal Power>}\end{aligned} \tag{242}$$

Therefore *misadjustment* from the MMSE solution is proportional to the LMS step size, the filter length, and the signal input power of $x(k)$. See also *Adaptive Signal Processing, Least Mean Squares Algorithm, Least Mean Squares Algorithm Convergence, Least Mean Squares Algorithm Time Constant, Wiener-Hopf Equations.*

Least Mean Squares (LMS) Algorithm Time Constant: The speed of convergence to a steady state error (expressed as an exponential time constant) can be precisely defined in terms of the eigenvalues of the correlation matrix, $\boldsymbol{R}$ (see *Least Mean Squares Algorithm Convergence*). A commonly used (if less accurate) measure is given by:

$$\tau_{mse} \approx \frac{N}{4\mu(\text{trace}[\boldsymbol{R}])} = \frac{1}{4\mu \text{<Signal Power>}} \tag{243}$$

Therefore the speed of adaption is proportional to the inverse of the signal power and the inverse of the step size. A large step size will adapt quickly but with a large MSE, whereas a small step size, will adapt slowly but achieve a small MSE. The design trade-off to select μ, is a requirement of the algorithm designer, and will, of course, depend of the particular application. See also *Adaptive Signal Processing, Adaptive Step Size, Least Mean Squares Algorithm, Least Mean Squares Algorithm Convergence, Least Mean Squares Algorithm Misadjustment, Wiener-Hopf Equations.*

Least Mean Squares (LMS) Algorithm Variants: A number of variants of the LMS exist. These variants can be split into three families: (1) algorithms derived to reduce the computation requirements compared to the standard LMS; (2) algorithms derived to improve the convergence properties over the standard LMS; (3) modifications of the LMS to allow a more efficient implementation.

In order to reduce computational requirements, the sign-error, sign-data and sign-sign LMS algorithms circumvent multiplies and replace them with shifting operations (which are essentially power of two multiplies or divides). The relevance of the sign variants of the standard LMS however is now somewhat dated due to the low cost availability of modern DSP processors where a multiply can be performed in the same time as a bit shift (and faster than multiple bit shifts). The convergence speed and achievable mean squared error for all of the sign variants of the LMS are less desirable than the for the standard LMS algorithm.

To improve convergence speed, the stability properties and ensure a small excess mean squared error the normalized, the leaky and the variable step size LMS algorithms have been developed. A summary of some of the LMS variants are:

- **Delay LMS:** The delay LMS simply delays the error signal in order that a “systolic” timed application specific circuit can be implemented:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu e(k-n)\boldsymbol{x}(k-n) \tag{244}$$

Note that the delay-LMS is in fact a special case of the more general filtered-X LMS.

- **Filtered-X LMS:** See *Least Mean Squares Filtered-X Algorithm*.
- **Filtered-U LMS:** See *Active Noise Control*.
- **Infinite Impulse Response (IIR) LMS:** See *Least Mean Squares - IIR Filter Algorithm*s.
- **Leaky LMS:** A leakage factor, *c*, can be introduced to improve the numerical behaviour of the standard LMS:

$$\boldsymbol{w}(k+1) = c\boldsymbol{w}(k) + 2\mu e(k)\boldsymbol{x}(k) \tag{245}$$

  By continually leaking the weight vector, $\boldsymbol{w}(k)$, even if the algorithm has found the minimum mean squared error solution it will require to continue adapting to compensate for the error introduced by the leakage factor. The advantage of the leakage is that the sensitivity to potentially destabilizing round off errors is reduced. In addition, in applications where the input occasionally becomes very small, *leaky LMS* drives the weights toward zero (this can be an advantage in noise cancelling applications). However the disadvantage to *leaky LMS* is that the achievable mean squared error is not as good as for the standard LMS. Typically *c* has a value between 0.9 (very leaky) and 1 (no leakage).

- **Multichannel LMS:** See [68]. .
- **Newton LMS:** This algorithm improves the convergence properties of the standard LMS. There is quite a high computational overhead to calculate the matrix vector product (and, possibly, the estimate of the correlation matrix $\boldsymbol{R}^{-1}$) at each iteration:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\boldsymbol{R}^{-1}\mu e(k)\boldsymbol{x}(k) \tag{246}$$

- **Normalised Step Size LMS:** The *normalised LMS* calculates an approximation of the signal input power at each iteration and uses this value to ensure that the step size is appropriate for rapid convergence. The normalized step size, μ_n, is therefore time varying. The normalised LMS is very useful in situations where the input signal power fluctuates rapidly and the input signal is slowly varying non-stationary:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu_n e(k)\boldsymbol{x}(k), \quad \mu_n = \frac{1}{\varepsilon + \|\boldsymbol{x}(k)\|^2} \tag{247}$$

  where ε is a small constant to ensure that in conditions of a zero input signal, $x(k)$, a divide by zero does not occur. $\|\boldsymbol{x}(k)\|$ is 2-norm of the vector $\boldsymbol{x}(k)$.

- **Sign Data/Regressor LMS:** The *sign data (or regressor) LMS* was first developed to reduce the number of multiplications required by the LMS. The step size, μ, is carefully chosen to be a power of two and only bit shifting multiplies are required:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu e(k)\text{sign}[\boldsymbol{x}(k)], \quad \text{sign}[x(k)] = \begin{cases} 1, & x(k) > 0 \\ 0, & x(k) = 0 \\ -1, & x(k) < 0 \end{cases} \tag{248}$$

- **Sign Error LMS:** The *sign error LMS* was first developed to reduce the number of multiplications required by the LMS. The step size, μ, is carefully chosen to be a power of two and only bit shifting multiplies are required:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu\text{sign}[e(k)]\boldsymbol{x}(k), \quad \text{sign}[e(k)] = \begin{cases} 1, & e(k) > 0 \\ 0, & e(k) = 0 \\ -1, & e(k) < 0 \end{cases} \tag{249}$$

- **Sign-SIgn LMS:** The *sign-sign error LMS* was first presented in 1966 to reduce the number of multiplications required by the LMS. The step size, μ, is carefully chosen to be a power of two and only bit shifting multiplies are required:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu \text{sign}[e(k)]\text{sign}[\boldsymbol{x}(k)], \qquad \text{sign}[z(k)] = \begin{cases} 1, & z(k) > 0 \\ 0, & z(k) = 0 \\ -1, & z(k) < 0 \end{cases} \tag{250}$$

- **Variable Step Size LMS:** The variable step size LMS was developed in order that when the LMS algorithm first starts to adapt, the step size is large and convergence is fast. However as the error reduces the step size is automatically decreased in magnitude in order that smaller steps can be taken to ensure that a small excess mean squared error is achieved:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu_v e(k)\boldsymbol{x}(k), \qquad \mu_v \propto E\{e^2(k)\} \tag{251}$$

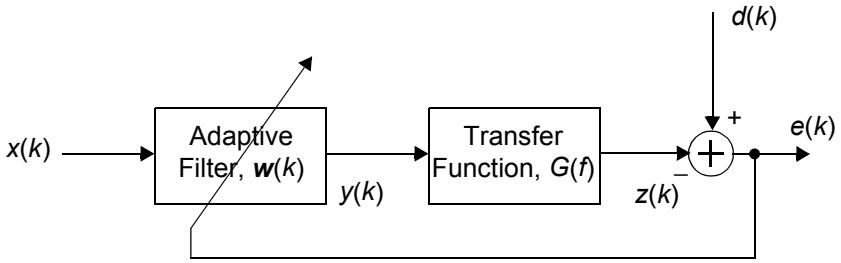
Alternatively variable step size algorithms can be set up with deterministic schedules for the modification of the step size. For example

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu_v e(k)\boldsymbol{x}(k), \qquad \mu_v = \mu 2^{-\text{int}(\lambda k)} \tag{252}$$

such that as time, k, passes the step size, μ_v, gets smaller in magnitude. μ is the step size calculated for the standard LMS, λ is a positive constant, and $\text{int}(\lambda k)$ is the closest integer to λk.

Note that a hybrid of more than one of the above LMS algorithm variants could also be implemented. See also *Adaptive Signal Processing, Least Mean Squares Algorithm, Least Mean Squares IIR Algorithms, Recursive Least Squares.*

Least Mean Squares (LMS) Filtered-X Algorithm: In certain control applications the adaptive architecture has a transfer function at the output of the adaptive filter:



This adaptive filtering architecture has a known transfer function at the output of the adaptive filter which filters $y(k)$ before subtraction from $d(k)$ to produce the error. Compare this to the generic adaptive filtering described previously (see *Adaptive Filtering*). Note that the DAC and ADC at the input and output respectively of the transfer function $G(f)$ are not shown for diagrammatic clarity.

In deriving the standard LMS algorithm the gradient of the instantaneous squared error was calculated. Note, however, in the above architecture the instantaneous error is given by:

$$\begin{aligned} e(k) &= d(k) - z(k) \\ &= d(k) - \{y(k) * g(k)\} \end{aligned} \tag{253}$$

where $g(k)$ is the perfectly sampled impulse response of the transfer function at the output of the adaptive filter, and the term $\{y(k) * g(k)\}$ is the result of $y(k)$ being convolved with $g(k)$. Therefore calculating the derivative of the instantaneous error produces:

$$\hat{\underline{\nabla}}_k = \frac{\partial}{\partial \boldsymbol{w}(k)} e^2(k) = 2e(k)\boldsymbol{f}(k) \tag{254}$$

where

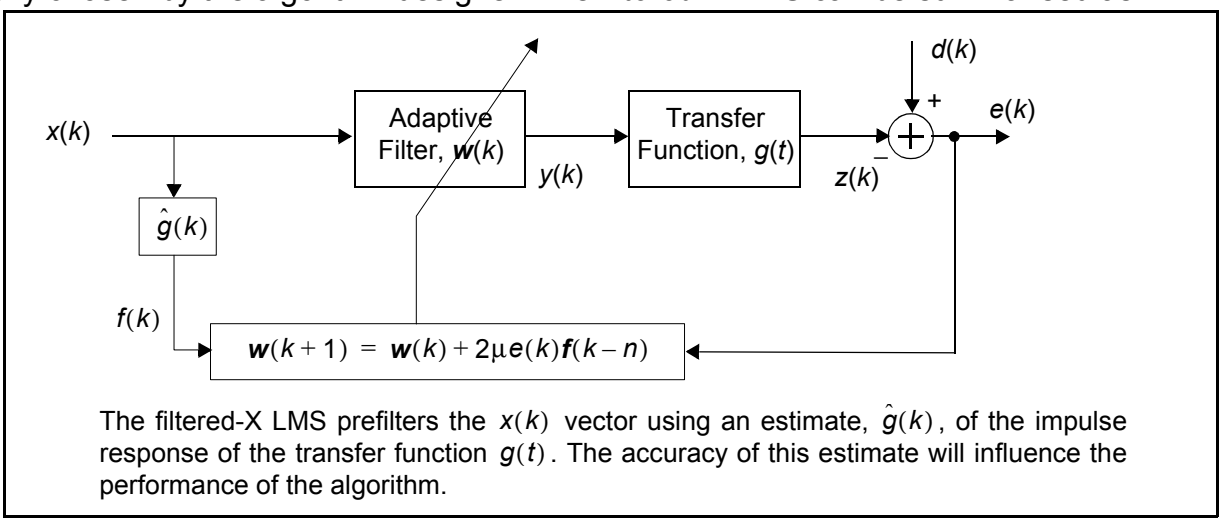
$$\boldsymbol{f}(k) = [f(k), f(k-1), f(k-2), \ldots, f(k-M+1)] \quad \text{and} \tag{255}$$

$$f(k) = \{x(k)^* g(k)\} \tag{256}$$

Therefore this algorithm requires that the pulse response $g(k)$ is known exactly in order to convolve with the input vector to create the $\boldsymbol{f}(k)$ vector. Clearly it is unlikely that $g(k)$ will be known exactly, however an estimate, $\hat{g}(k)$ can be found by apriori system identification. Therefore the *filtered-X LMS* algorithm is:

$$\begin{aligned} \boldsymbol{w}(k+1) &= \boldsymbol{w}(k) + 2\mu e(k)\boldsymbol{f}(k-n) \\ f(k) &= \sum_{n=0}^{M-1} \hat{g}(k)x(k-n) \end{aligned} \tag{257}$$

where M is the number of filter weights used in the FIR filter estimate of $g(k)$. Note that the number of weights in this estimate will influence the performance of the algorithm; too few weights may not adequately model the transfer function and could degrade performance. Therefore M must be carefully chosen by the algorithm designer. The *filtered-X LMS* can be summarised as:

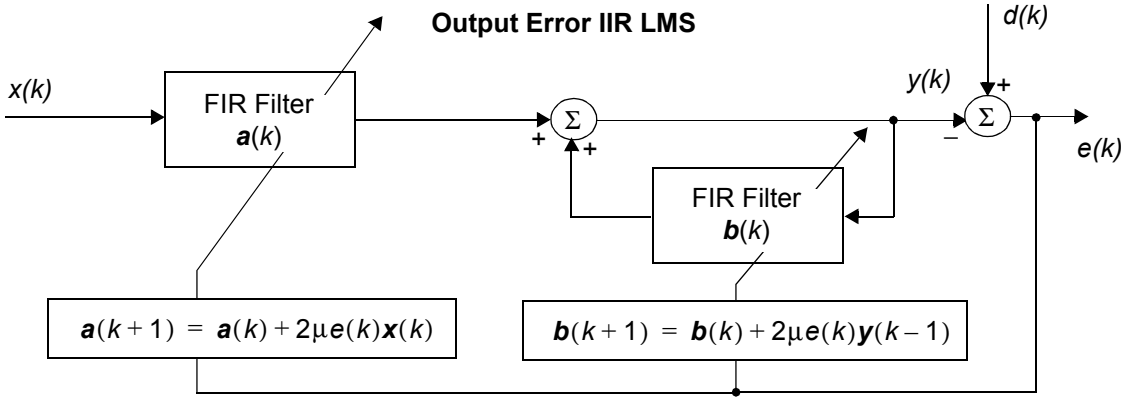


The filtered-X LMS prefilters the $x(k)$ vector using an estimate, $\hat{g}(k)$, of the impulse response of the transfer function $g(t)$. The accuracy of this estimate will influence the performance of the algorithm.

See also *Active Noise Control, Adaptive Signal Processing, Adaptive Step Size, Inverse System Identification, Least Mean Squares (LMS) Algorithm.*

Least Mean Squares (LMS) IIR Algorithms: Recently adaptive filtering algorithms based on IIR filters have been investigated for a number of applications. A good overview of adaptive IIR filters can be found in [36], [132]. The very simplest form of adaptive IIR LMS, sometimes referred to as Feintuch's algorithm [71], can be represented as:

In addition to the normal step size stability concerns of adaptive filters, the adaptive IIR LMS filter instability can also result if the poles of the filter migrate outside of the unit circle. Therefore extreme

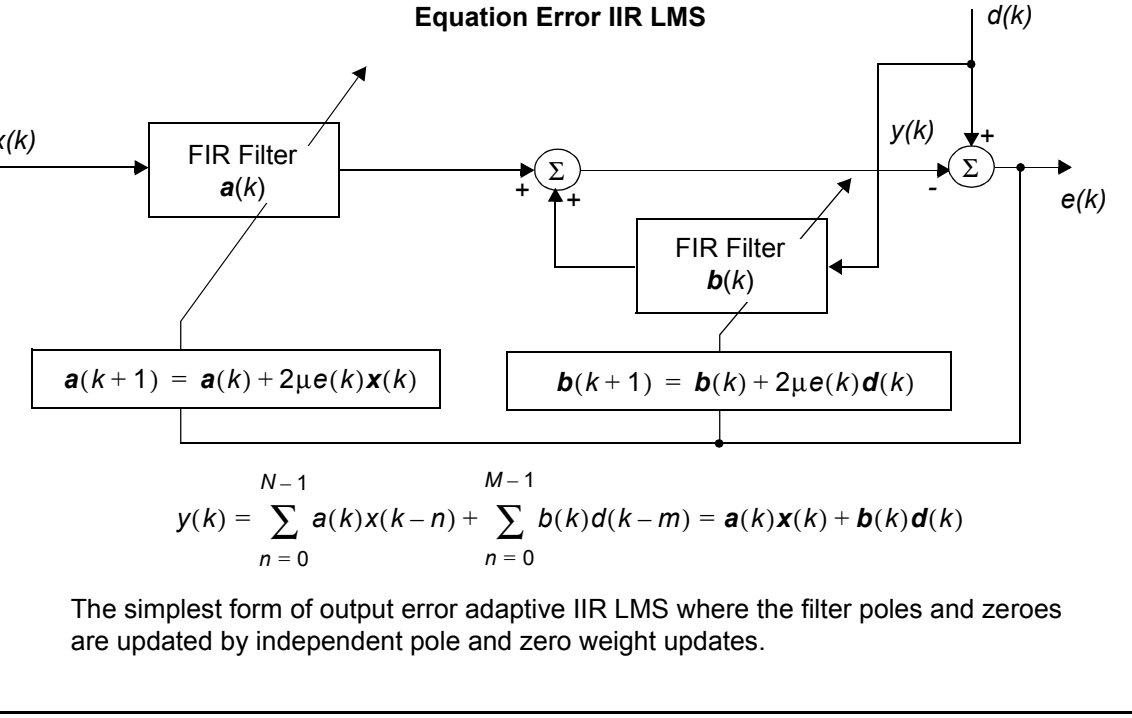


$$y(k) = \sum_{n=0}^{N-1} a(k)x(k-n) + \sum_{n=1}^{M-1} b(k)y(k-m) = \boldsymbol{a}(k)\boldsymbol{x}(k) + \boldsymbol{b}(k)\boldsymbol{y}(k-1)$$

The simplest form of output error adaptive IIR LMS where the filter poles and zeroes are updated by independent pole and zero weight updates.

care is necessary when choosing the adaptive step size for both recursive and non-recursive weight updates. While this simple (some would say simple-minded) algorithm appears to be useless, it is surprisingly robust in a wide variety of applications.

In order to address the problem of poles migrating outside of the unit circle, one suggestion has been the equation error adaptive IIR LMS filter which is actually the updating of two independent FIR filters:



$$y(k) = \sum_{n=0}^{N-1} a(k)x(k-n) + \sum_{n=0}^{M-1} b(k)d(k-m) = \boldsymbol{a}(k)\boldsymbol{x}(k) + \boldsymbol{b}(k)\boldsymbol{d}(k)$$

The simplest form of output error adaptive IIR LMS where the filter poles and zeroes are updated by independent pole and zero weight updates.

In conditions of high observation noise the equation error will give a biased (and very poor!) solution. See also *Active Noise Control, Adaptive Signal Processing, Least Mean Squares Algorithm.*

Least Significant Bit (LSB): The bit in a binary number with the least arithmetic numerical significance. See also *Most Significant Bit, Sign Bit.*

MSB **LSB**

-128	64	32	16	8	4	2	1
0	1	0	1	1	0	1	1

In 2's complement notation the MSB has a negative weighting.

$= 64 + 16 + 8 + 2 + 1 = 91_{10}$

Least Squares: Given the overdetermined linear set of equations, $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A}$ is a known $m \times n$ matrix of rank n (with $m > n$), $\boldsymbol{b}$ is a known m element vector, and $\boldsymbol{x}$ is an unknown n element vector, then the *least squares* solution is given by:

$$\boldsymbol{x}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{258}$$

(Note that if the problem is underdetermined, $m < n$, then Eq.258 is not the solution, and in fact there is no unique solution; a good (i.e., close) solution can often be found however using the pseudoinverse obtained via singular value decomposition.)

The *least squares* solution can be derived as follows. Consider again the overdetermined linear set of equations:

$$\underbrace{\begin{bmatrix} a_1 & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ a_{41} & a_{42} & \dots & a_{4n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\boldsymbol{x}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_m \end{bmatrix}}_{\boldsymbol{b}} \tag{259}$$

If $\boldsymbol{A}$ is a nonsingular square matrix, i.e. $m = n$, then the solution can be calculated as:

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{260}$$

However if $m \neq n$ then $\boldsymbol{A}$ is a rectangular matrix and therefore not invertible, and the above equation cannot be solved to give exact solution for $\boldsymbol{x}$. If $m < n$ then the system is often referred to as underdetermined and an infinite number of solutions exist for $\boldsymbol{x}$ (as long as the m equations are consistent). If $m > n$ then the system of equations is overdetermined and we can look for a solution by striving to make $\boldsymbol{Ax}$ be as close as possible to $\boldsymbol{b}$, by minimizing $\boldsymbol{Ax} - \boldsymbol{b}$ in some sense. The most mathematical tractable way to do this is by the method of *least squares*, performed by minimizing the 2-norm denoted by e :

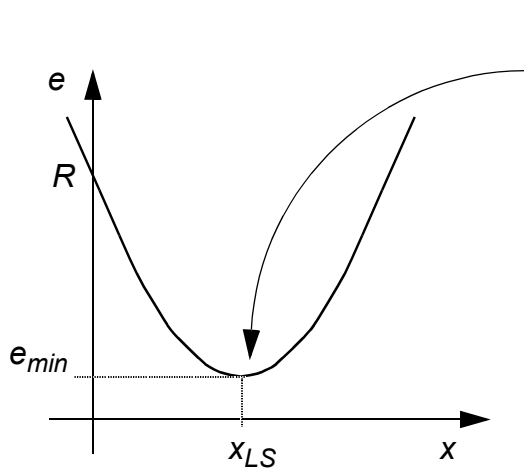
$$e = (\|\boldsymbol{Ax} - \boldsymbol{b}\|_2)^2 = (\boldsymbol{Ax} - \boldsymbol{b})^T(\boldsymbol{Ax} - \boldsymbol{b}) \tag{261}$$

Plotting e against the n-dimensional vector $\boldsymbol{x}$ gives a hyperparabolic surface in (n+1)-dimensions. . If $n = 1$, $\boldsymbol{x}$ has only one element and the surface is a simple parabola. For example consider the case where $\boldsymbol{A}$ is a 2×1 matrix, then from Eq. 261:

$$
\begin{aligned}
e &= \left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} x - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}\right)^T \left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} x - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}\right) \\
&= \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} x^2 - 2 \begin{bmatrix} b_1 & b_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} x + \begin{bmatrix} b_1 & b_2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \\
&= \left[a_1^2 + a_2^2\right] x^2 - 2\left[a_1 b_1 + a_2 b_2\right] x + \left[b_1^2 + b_2^2\right] \\
&= Px^2 - Qx + R
\end{aligned}
\tag{262}
$$

where $P = \left[a_1^2 + a_2^2\right]$, $Q = 2\left[a_1 b_1 + a_2 b_2\right]$ and $R = \left[b_1^2 + b_2^2\right]$.

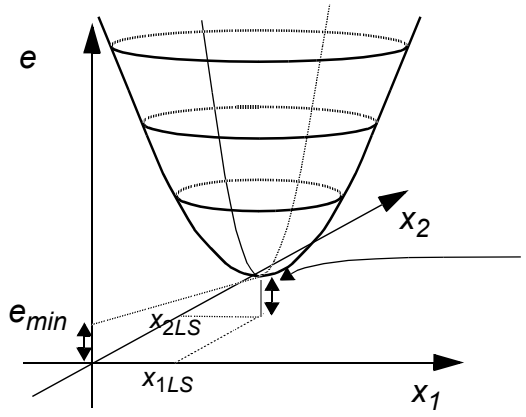
Clearly the minimum point on the surface lies at the bottom of the parabola:



$$\frac{de}{dx} = 2Px_{LS} - Q = 0 \quad \Rightarrow \quad x_{LS} = \frac{Q}{2P}$$

$$x_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} = \frac{\left[a_1 b_1 + a_2 b_2\right]}{\left[a_1^2 + a_2^2\right]} = \frac{Q}{2P}$$

If n = 2, $\boldsymbol{x} = [x_1 \;\; x_2]^T$ and the error surface is a paraboloid. This surface has one minimum point at the bottom of the paraboloid where the gradient of the surface with respect to both x_1 and x_2 axis:



$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\frac{de}{d\boldsymbol{x}} = \begin{bmatrix} \frac{de}{dx_1} \\ \frac{de}{dx_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

If the $\boldsymbol{x}$ vector has three or more elements ($n \geq 3$) the surface will be in four or more dimensions and cannot be shown diagrammatically.

To find the minimum value of e for the general case of an n-element $\boldsymbol{x}$ vector the "bottom" of the hyperparaboloid can be found by finding the point where the gradient in every dimension is zero (cf. the above 1 and 2-dimensioned examples). Therefore differentiating e with respect to the vector $\boldsymbol{x}$:

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{x}} = \left[\frac{\mathrm{d}e}{\mathrm{d}x_1}\ \frac{\mathrm{d}e}{\mathrm{d}x_2}\ \frac{\mathrm{d}e}{\mathrm{d}x_3}\ \cdots\ \frac{\mathrm{d}e}{\mathrm{d}x_n}\right]^T = 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \tag{263}$$

and setting the gradient vector to the zero vector,

$$\frac{\mathrm{d}e}{\mathrm{d}\boldsymbol{x}} = \boldsymbol{0} \tag{264}$$

to find the minimum point, e_{min}, on the surface gives the least squares error solution for $\boldsymbol{x}_{LS}$:

$$\begin{aligned} 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}_{LS} - \boldsymbol{b}) &= \boldsymbol{0} \\ \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_{LS} - \boldsymbol{A}^T\boldsymbol{b} &= \boldsymbol{0} \\ \boldsymbol{x}_{LS} &= (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \end{aligned} \tag{265}$$

If the rank of matrix $\boldsymbol{A}$ is less than n, then the inverse matrix $(\boldsymbol{A}^T\boldsymbol{A})^{-1}$ does not exist and the least squares solution cannot be found using Eq. 265 and the pseudoinverse requires to be calculated using singular value decomposition techniques. Note that if $\boldsymbol{A}$ is an invertible square matrix, then the *least squares* solution simplifies to:

$$\boldsymbol{x} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{A}^{-T}\boldsymbol{A}^T\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{266}$$

See also *Matrix Decompositions - Singular Value Decompositions, Matrix Inversion, Minimum Residual, Normal Equations, Least Mean Squares, Least Squares Residual, Square System of Equations, Overdetermined System, Recursive Least Squares*.

Least Squares Residual: The least squares error solution to the overdetermined system of equations, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, is given by:

$$\boldsymbol{x}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{267}$$

where $\boldsymbol{A}$ is a known $m \times n$ matrix of rank n and with $m > n$, $\boldsymbol{b}$ is a known m element vector, and $\boldsymbol{x}$ is an unknown n element vector. The *least squares residual* given by:

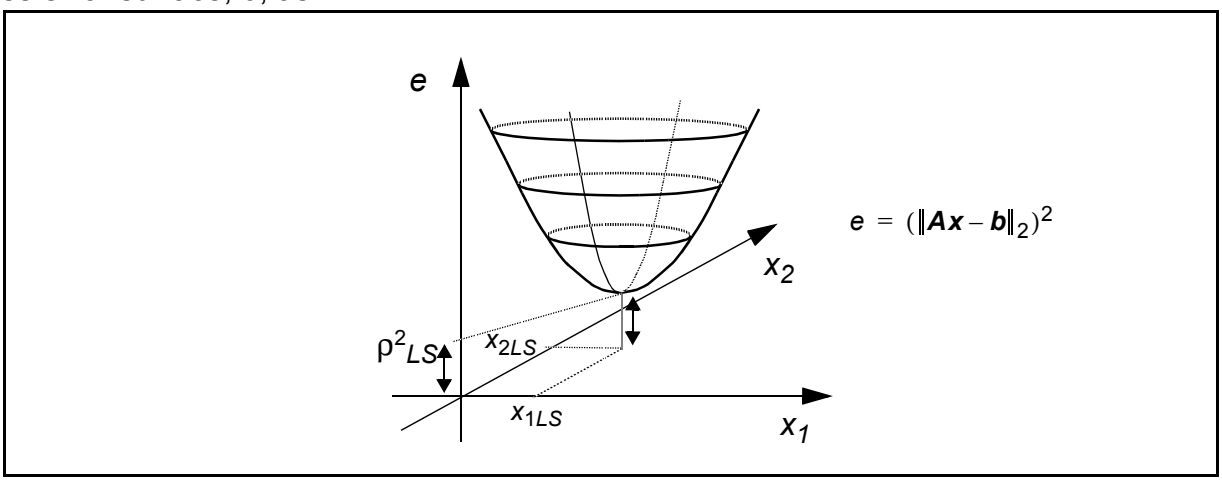
$$\boldsymbol{r}_{LS} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{LS} \tag{268}$$

is a measure of the error obtained when using the method of least squares. The smaller the value of $\boldsymbol{r}_{\mathrm{LS}}$, then the more accurately $\boldsymbol{b}$ can be generated from the columns of the matrix $\boldsymbol{A}$. The

magnitude or size of the *least squares residual* is calculated by finding the squared magnitude, or 2-norm of $\boldsymbol{r}_{LS}$:

$$\rho_{LS} = \|\boldsymbol{A}\boldsymbol{x}_{LS} - \boldsymbol{b}\|_2 \tag{269}$$

As an example, for a system with $n = 2$ the *least squares residual* can be shown on the least squares error surface, e, as:



Note that if $m = n$, and A is a non-singular matrix, then $\rho_{LS} = 0$. See also *Least Squares*, *Matrix, QR Algorithm*, *Recursive Least Squares*.

L_{eq}: See *Equivalent continuous level.*

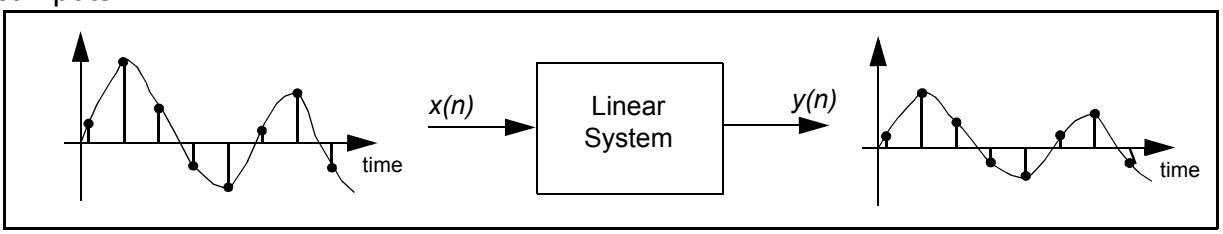
Linear Algebra: *Linear algebra* is an older branch of mathematics that uses matrix based equations. The computer has spawned a rebirth of interest in linear algebra and changed what was thought to be an arcane, obsolete and strictly academic area into a ubiquitous, fundamental tool in virtually every applied, pure or social science field. Over the last few years the advent of fast DSP processors has led to the solution of many DSP problems using numerical *linear algebra* [15]. See also *Matrix, Matrix Algorithms, Matrix Decompositions, Matrix Properties.*

Linear Feedback Shift Register (LFSR): A simple shift register with feedback and combinational logic using for the generation of pseudo random binary noise. See *Pseudo Random Binary Sequence*.

Linear Phase Filter: See *Finite Impulse Response Filter*.

Linear Predictive Coding (LPC): *Linear predictive coding* is a compression algorithm for reducing the storage requirements of digitized speech. In *LPC* the vocal tract is modelled as an all-pole digital filter (IIR) and the calculated filter coefficients are used to code the speech down to levels of 2400 bits/sec from speech sampled at 8kHz with 8 bit resolution.

Linear System: A system is said to be linear if the weighted sum of the system output given two distinct inputs equals the system output given a single input equal to the weighted sum of the two distinct inputs.:



In general, for a *linear system* $y(n) = f(x(n))$, if, whenever:

$$\begin{aligned} y_1(n) &= f[x_1(n)] \\ y_2(n) &= f[x_2(n)] \end{aligned} \tag{270}$$

then:

$$a_1 y_1(n) + a_2 y_2(n) = f[a_1 x_1(n) + a_2 x_2(n)] \tag{271}$$

for all values of a_1 and a_2. For example consider the linear system:

$$y(n) = 4.3x(n) + 6.01x(n-1) \tag{272}$$

If $x_1(n)$ = sin100nt, then the output which will be denoted as y_1(n), is given by:

$$y_1(n) = 4.3\sin 100nt + 6.01\sin 100(n-1)t \tag{273}$$

For a different input $x_2(n)$ = sin250nt, then the output denoted as y_2(n) is given by:

$$y_2(n) = 4.3\sin 250nt + 6.01\sin 250(n-1)t \tag{274}$$

Therefore, given that the system is linear, if $x_3(n)$ = sin100nt + sin250nt, then:

$$\begin{aligned} y_3(n) &= 4.3(\sin 100nt + \sin 250nt) + 6.01(\sin 100(n-1)t + \sin 250(n-1)t) \\ &= y_1(n) + y_2(n) \end{aligned} \tag{275}$$

In general inputting a sine wave to a *linear system* will yield an output that is a sine wave at exactly the same frequency but with modified phase and magnitude. If any other frequencies are output (e.g., if the sine wave is distorted in any way) then the system is nonlinear. (Note that this is not true for other waveforms; inputting a square to a *linear system* is unlikely to produce a square wave at the output. If the square wave is viewed as its sine wave components (from Fourier analysis) then the output of the *linear system* should only contain sine waves at those frequencies, but where the modification of their amplitude, phase and frequency means that their superposition no longer gives a square wave.)

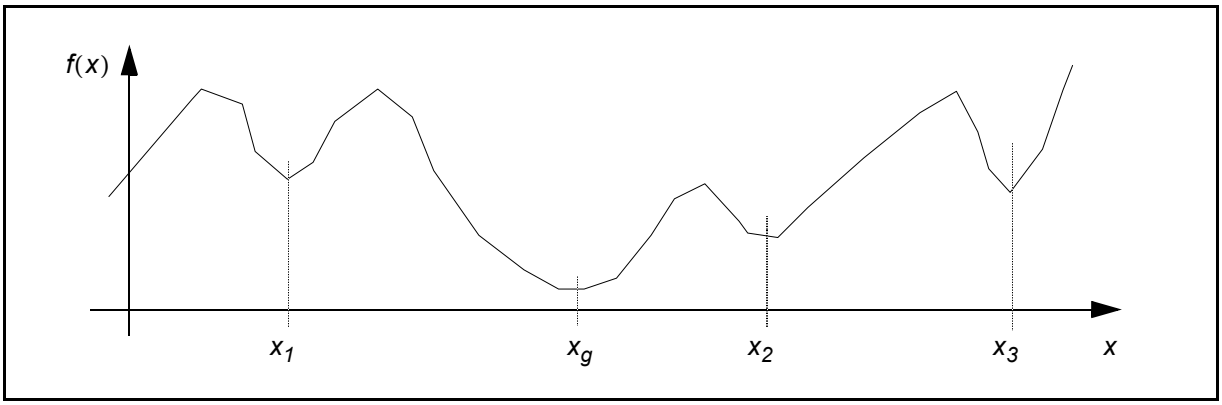
DSP systems such as digital filters (IIR and FIR) are linear filters. Any filter that has time varying weights, however, is non-linear. See also *Distortion, Non-Linear System, Poles, Transfer Function, Frequency Response*.

Linearity: *Linearity* is the property possessed by any system which is linear.

Linearly Dependent Vectors: See *Vector Properties and Definitions - Linearly Dependent.*

LL^T: See *Matrix Decompositions - Cholesky Decomposition*.

Local Minima: The global minimum is the smallest value taken on by that function. For example for the function, $f(x)$, the global minimum is at $x = x_g$. The minima are x_1, x_2 and x_3 are termed *local minima*:



When attempting to use least squares, or least mean squared based algorithms to find the global minimum of a function, the zero gradient of the function is found. For a quadratic surface with only one minimum the method works very well. However if the surface in not quadratic, then the solution obtained is not necessarily the global minimum, as the gradient is also zero at the *local minima* (and the local maxima and inflection points). See also *Adaptive IIR Filters, Hyperparaboloid, Global Minima, Least Squares, Simulated Annealing*.

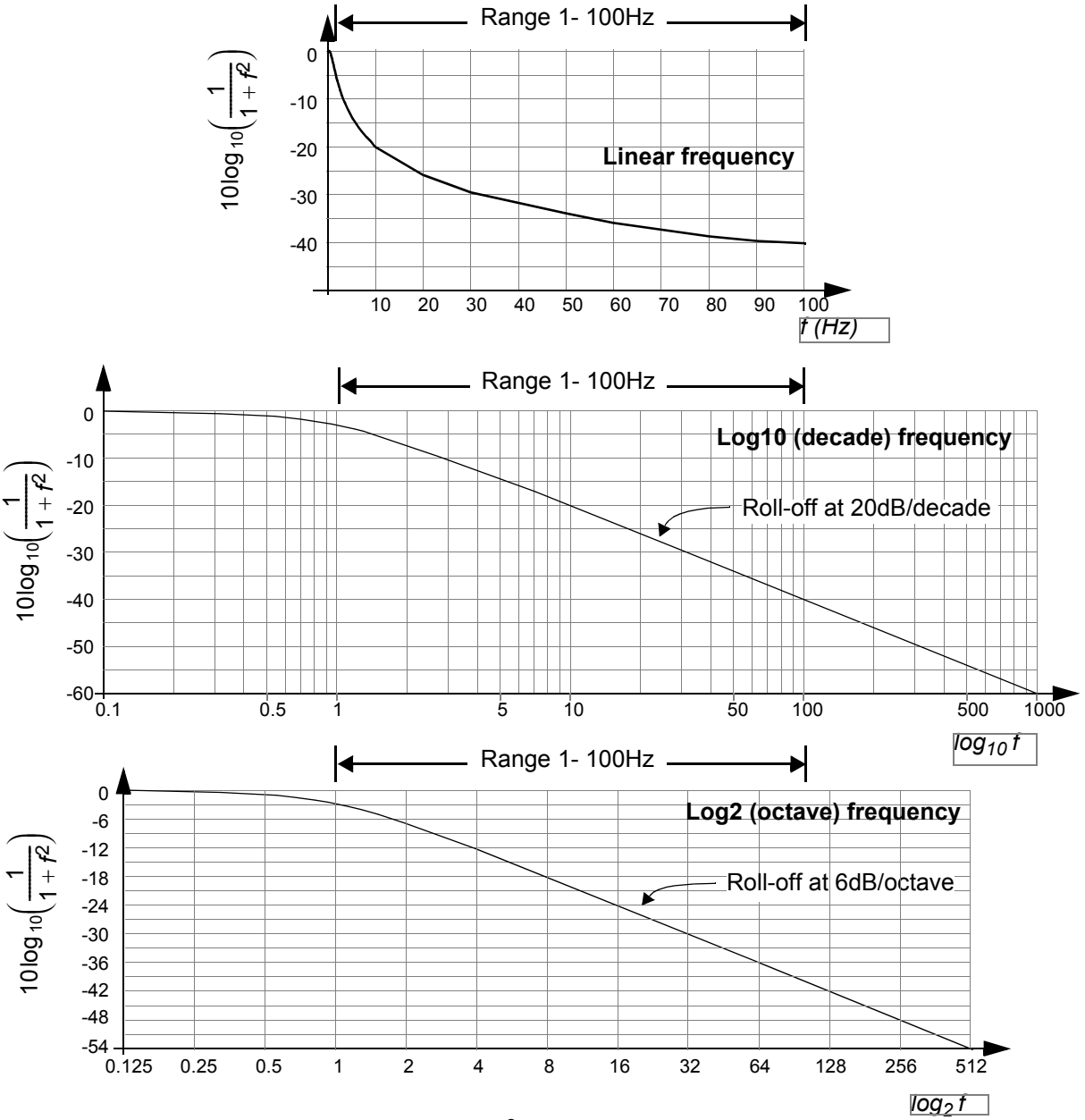
Localization: When used in the context of acoustics, *localization* is the ability to perceive the direction from which sounds are coming. For animals the two ears provide excellent instruments of *localization*. *Localization* problems are also found in radar and sonar systems where arrays of sensors are used to sense the direction from which signals are radiating. Generally, a minimum of two sensors are required to accurately localize a sound source. A current focus of research is in producing arrays of microphones using DSP algorithms to improve sound quality for applications such as hands-free telephony, hearing aids, and concert hall microphone pick-ups. Some applications require that a desired source be located before it can be extracted or filtered from the rest of the sound field. See also *Audiology, Beamforming, Lateralization.*

Logarithmic Amplitude: If the amplitude range of a signal or system is very large then it is often convenient to plot the magnitude on a logarithmic scale rather than a linear scale. The most common form of logarithmic magnitude uses the logarithmic decibel scale which represents a ratio of two powers. See also *Decibels (dB)*.

Logarithmic Frequency: When the frequency range of a signal or system is very large, it is often convenient to plot the frequency axis on a logarithmic rather than a linear scale. The human ear, for example, has a sensitivity range from around 70Hz to 15000Hz and is often described as being a *logarithmic frequency response*. Logarithmically spaced frequencies are equally spaced distances on the basilar membrane within the cochlea. The perception of frequency change is such that a doubling of frequency from 200Hz to 400Hz is perceived as being the same change as a doubling of frequency from 2000Hz to 4000Hz, i.e., both sounds have increased by an octave. In DSP

systems everything from digital filter frequency responses, to spectrograms may be represented with a *logarithmic frequency* scale. See also *Wavelet Transform*.

The most common logarithmic scales are *decade* ($\log_{10}f$) and *octave* ($\log_2 f$) although clearly any logarithmic base can be used. If the y-axis is also plotted on a logarithmic scale (such as dB), then the graph is *log-log*. See also *Decibels, Roll-off .*



Graphs of the second order system $1/(1+f^2)$. The range of 1 to 100Hz is the width on all three graphs. Clearly using a logarithmic scale allows much greater frequency ranges to be represented than with a linear scale. More resolution is available at the lower frequencies (0 to 1 Hz), although at higher frequencies there is less resolution.

Lossless Compression: If a compression algorithm is lossless, then the signal information (or entropy) after the signal has been compressed and decompressed has not changed, i.e. all signal information has been retained. Hence, the uncompressed signal is identical to the original signal.

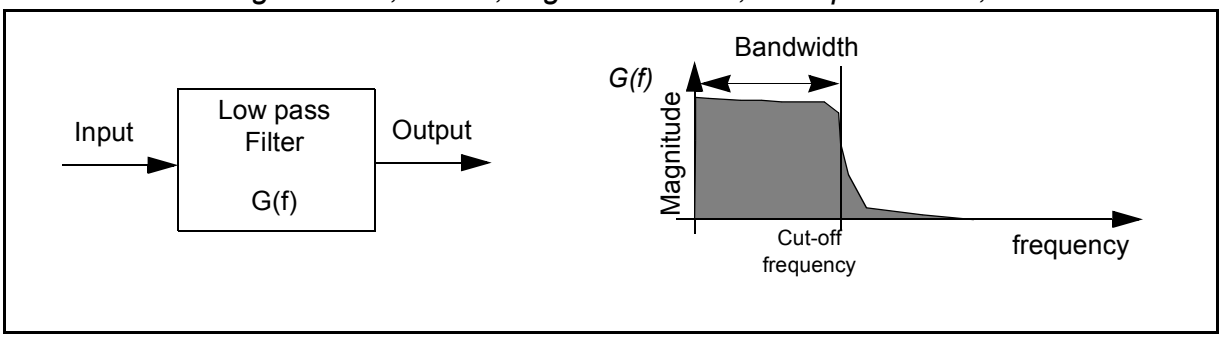
Lossless compression for digital audio signals is not particularly successful and is likely to achieve at best 2.5:1 compression ratio [61]. See also *Compression, Lossy Compression*.

Lossy Compression: If a compression algorithm is *lossy*, then the signal information (or entropy) after the signal has been compressed and decompressed is reduced, i.e. some signal information has been lost. However if the *lossy* algorithm is carefully designed then the elements of the signal that are lost are not particularly important to the integrity of the signal. For example, the precision adaptive subband coding (PASC) algorithm compresses a hifidelity digital audio signal by a factor of 4, however the information that is "lost" would not have been perceived by the listener due to masking effects of the human ear. Alternatively if very high levels of compression are being attempted then the *lossy* effects of the algorithm may be quite noticeable. See also *Compression, Lossless Compression*.

Loudness Recruitment: Defects in the auditory mechanism can lead to a hearing impairment whereby the dynamic range from the threshold of audibility to the threshold of discomfort is greatly reduced [30]. *Loudness recruitment* is the abnormally rapid growth in perceived loudness (versus intensity) in individuals with reduced dynamic range of audibility. The range of hearing is nominally 120dB(SPL). However, in persons with hearing loss, the range may be as low as 40dB. These individuals have a raised threshold of audibility, but after sounds exceed that threshold the perceived loudness grows rapidly until they reach normal perceived loudness for sounds near the threshold of discomfort. This growth in their perceived loudness is termed *loudness recruitment*. One common misconception is that individuals with *loudness recruitment* are more sensitive to changes in intensity (i.e., they have smaller intensity JNDs or DLs). When tested, however, their JNDs for intensity are very near normal -- this indicates that they have fewer different perceptible difference limens (DLs) over the normal range of loudness than normal hearing individuals. See also *Ear, Equal Loudness Contours, Hearing Aids, Threshold of Hearing*.

Low Noise Components: All electronic components introduce certain levels of unwanted and potentially interfering noise. *Low noise components* introduce lower levels of noise than standard components, but the cost is usually higher.

Low Pass Filter: A filter which passes only the portions of a signal that have frequencies between DC (0 Hz) and a specified cut-off frequency. Frequencies above the cut-off frequency are highly attenuated. See also *Digital Filter, Filters, High Pass Filter, Bandpass Filter, Filters.*



Lower Triangular Matrix: See *Matrix Structured - Lower Triangular.*

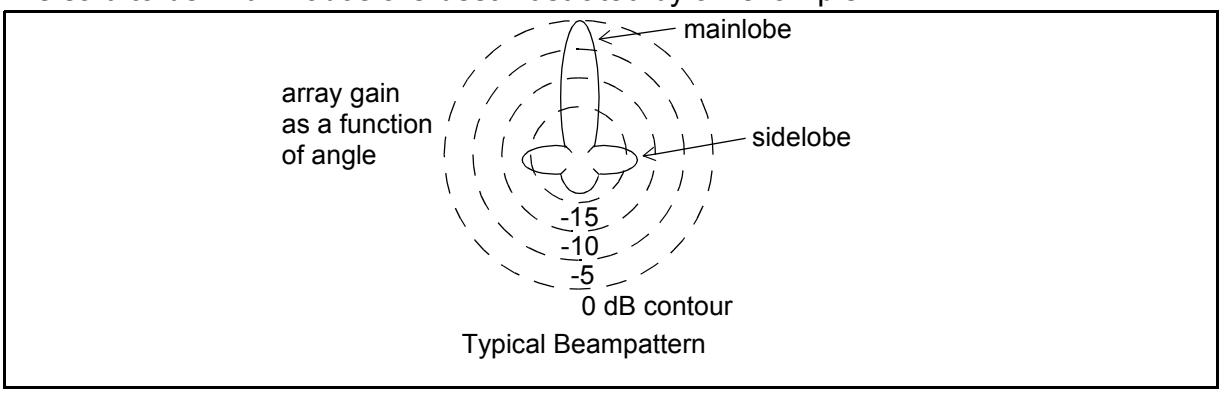
LU: See *Matrix Decompositions - LU Decomposition*.

M

m-sequences: Shorthand term for a maximum length sequence. See *Maximum Length Sequences, Pseudo-Random Binary Sequence*.

Machine Code: The binary codes that are stored in memory and are fetched by the DSP processor to be executed on the chip and perform some useful function, such as multiplication of two numbers. Collectively *machine code* instructions form a meaningful program. *Machine code* is usually generated (by the assembler program) from source code written in the assembly language. This machine code can then be downloaded onto the DSP processor. *Machine code* has a one to one correspondence with assembly language. See also *Assembly Language, Cross Compiler.*

Main Lobe: In an antenna or sensor array processing system, *main lobe* refers to the primary lobe of sensitivity in the beampattern. For a filter or a data window, *main lobe* refers to the primary passband lobe of sensitivity. The more narrow the *main lobe*, the more selective or sensitive a given system is said to be. *Main lobes* are best illustrated by an example.



See also *Beamformer, Beampattern, Sidelobes, Windows.*

Magnitude Response: See *Fourier Series - Complex Exponential Representation*.

Mammals: While not using digital signal processing capabilities, many mammals do of course use analog signals for communication and navigation. Most obviously *mammals* (including humans) use acoustic signals for communication via, for example, speech (humans), barking (dogs), and so on. Elephants communicate with very low frequencies (around 100Hz and well below -- even down to a few Hz), and can therefore communicate over very long distances via acoustic waves travelling in the ground. These ground-borne waves suffer less attenuation than airborne acoustic waves. It was this low frequency rumble communication that caused many early elephant watchers to believe that elephants had ESP (extra sensory perception) ability. Light signals (from the electromagnetic family) are used by most animals for navigation and communication purposes. Another well known use of signal processing is by the bat which uses sonar blips to avoid objects in its path during night flying. The magnetic field sensing abilities of birds and bees is another well known though not fully understood use of signal processing for navigation. Some *mammals* (mainly antipodean), such as the platypus an the echidna have electroreception abilities. See also *Electroreception*.

Marginally Stable: If a discrete system has poles on the unit circle then it can be described as marginally stable. See *Dual Tone Multifrequency*.

Masking: *Masking* refers to the process whereby one particular sound is close to inaudible in the presence of another louder signal. *Masking* is more precisely defined as spectral or temporal, although in audio and speech coding the term is usually used in reference to spectral *masking*. For spectral *masking* a loud signal raises the threshold of hearing of signals of a lower level but with slightly higher or lower frequencies. This effectively leaves these other signals inaudible. For temporal *masking*, sounds that occur a short time before of after a louder sound are not perceived. Simultaneous *masking* is also used in audiometry in order to minimize the perceivable conductance of test tones from the ear under test by injecting noise into the ear not being tested. See also *Audiometry, Spectral Masking, Temporal Masking, Threshold of Hearing*.

Masking Pattern Adapted Universal Subband Integrated Coding and Multiplexing (MUSICAM): MUSICAM was developed jointly by CCETT (France), IRT (Germany) and Philips (the Netherlands), amongst others, originally for the application of digital audio broadcasting (DAB). MUSICAM is based on subband psychoacoustic compression techniques and has been incorporated into MPEG-1 in combination with the ASPEC compression system. See also *Adaptive Spectral Perceptual Entropy Coding (ASPEC), Precision Adaptive Subband Coding (PASC), Psychoacoustics, Spectral Masking, Temporal Masking*.

Matlab: A program produced by the MathWorks that allows high level simulation of matrix and DSP systems, with excellent post-processing graphics facilities for data presentation. Libraries containing virtually every DSP operation are widely available for *Matlab*.

Matrix: A *matrix* is a set of numbers stored in a 2 dimensional array usually to represent data in an ordered structure. If $\Re$ denotes the set of real numbers, then the vector space of all $m \times n$ real *matrices* is denoted by $\Re^{m \times n}$, and if

$$\boldsymbol{A} \in \Re^{m \times n} \quad \text{then} \quad \boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \quad \text{with} \quad a_{ij} \in \Re, \quad \text{for} \begin{cases} 0 \le i \le m \\ 0 \le j \le n \end{cases} \tag{276}$$

where the symbol $\in$ simply means "is an element of" -- so $\boldsymbol{A}$ is an $m \times n$ *matrix*. The ordering of the data values is important to the information being conveyed by the *matrix*. The dimensions of a *matrix* are specified as the number of rows by the number of columns (the rows running from left to right, and the columns from top to bottom). *Matrices* are *usually* denoted in upper-case boldface font or upper case font with an underscore, e.g. $\boldsymbol{M}$ or $\underline{M}$. (Note that vectors are usually represented in lower case boldface font or lower font with an underscore, e.g. $\boldsymbol{v}$ or $\underline{v}$.

As an example a particular 4×3 *matrix*, $\boldsymbol{A}$, is:

$$\boldsymbol{A} = \begin{bmatrix} 4 & 9 & 2 \\ 10 & 1 & 13 \\ 3 & 4 & 5 \\ 1 & 2 & 2 \end{bmatrix} \quad \text{a 4 (row) by 3 (column) matrix} \tag{277}$$

Clearly each element in the *matrix* can be denoted by a subscript which refers to its row and column position:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \tag{278}$$

In the example, $a_{12} = 9$, and $a_{32} = 4$.

In DSP algorithms and analysis, matrices are extremely useful for compact and convenient mathematical representation of data and algorithms. For example the Wiener Hopf solution, and the Recursive Least Squares algorithm are expressed using matrix equations. See also *Matrix Algorithms, Matrix - Complex, Matrix Decompositions, Matrix Identities, Matrix Properties, Vector*.

Matrix - Complex: Each element in an $m \times n$ *complex matrix* is a complex number. The complex vector space is often denoted as $\boldsymbol{C}^{m \times n}$ where every element of that space is a complex number $c_{ij} \in \boldsymbol{C}$. Scaling, addition, subtracting and multiplication of a *complex matrix* is performed in the same way as for real matrices, except that the arithmetic is complex. For example:

$$\boldsymbol{Cd} + \boldsymbol{a} = \begin{bmatrix} 1+2j & 2+j \\ 3 & 2-0.5j \end{bmatrix} \begin{bmatrix} 1-3j \\ -2j \end{bmatrix} + \begin{bmatrix} 4-4j \\ 1+3j \end{bmatrix} = \begin{bmatrix} 9-5j \\ 2-13j \end{bmatrix} + \begin{bmatrix} 4-4j \\ 1+3j \end{bmatrix} = \begin{bmatrix} 13-9j \\ 3-10j \end{bmatrix} \tag{279}$$

Simple row column transposition (i.e. transpose operation) of complex matrices is not normally performed, but instead the Hermitian transpose is done where the matrix is transposed in the normal row-column style, but every element is complex conjugated. In DSP applications such as beamforming and digital communications, complex representation of information is often used for convenience. See also *Matrix, Matrix Properties - Hermitian Transpose*.

Matrix Algorithms: There are a number of well known *matrix algorithms* used in DSP for solving structured systems of equations. These algorithms are invariably used after a suitable decomposition has been performed on a matrix in order to produce a structured matrix/system of equations. See also *Matrix, Matrix Decompositions, Matrix - Partitioning*.

- **Back Substitution:** If an upper triangular system of linear equations:

$$\boldsymbol{Ux} = \boldsymbol{b} \quad \Rightarrow \quad \begin{bmatrix} u_{11} & \dots & u_{1,n-2} & u_{1,n-1} & u_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & u_{n-2,n-2} & u_{n-2,n-1} & u_{n-2,n} \\ 0 & \dots & 0 & u_{n-1,n-1} & u_{n-1,n} \\ 0 & \dots & 0 & 0 & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \\ b_n \end{bmatrix} \tag{280}$$

has to be solved for the unknown n element vector $\boldsymbol{x}$, where $\boldsymbol{U}$ is an $n \times n$ non-singular upper triangular matrix, then the last element of the unknown vector, x_n can be calculated from multiplication of the last row of $\boldsymbol{U}$ with the vector $\boldsymbol{x}$:

$$u_{nn} x_n = b_n \quad \Rightarrow \quad x_n = \frac{b_n}{u_{nn}} \tag{281}$$

the second last element can therefore be calculated from multiplication of the second last row of $\boldsymbol{U}$ with vector $\boldsymbol{x}$, and substitution of x_n from Eq. 281:

$$u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = b_{n-1} \quad \Rightarrow \quad x_{n-1} = \frac{b_{n-1} - u_{n-1,n}\left(\frac{b_n}{u_{nn}}\right)}{u_{n-1,n-1}} \tag{282}$$

In general it can be shown that all elements of $\boldsymbol{x}$ can be calculated recursively from:

$$x_i = \frac{b_i - \sum_{j=i+1} u_{ij}x_j}{u_{ii}} \tag{283}$$

This method of solving an upper triangular system of linear equations is called backsubstitution. Note that if the diagonal elements of $\boldsymbol{U}$ are very small relative to the off-diagonal elements, then the arithmetic required for the computation may require a large dynamic range. See also Matrix Decompositions - Cholesky/Forward Substitution/Gaussian Elimination/QR.

- **Forward Substitution:** If a system of lower triangular linear equations:

$$\boldsymbol{Lx} = \boldsymbol{b} \quad \Rightarrow \quad \begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} \tag{284}$$

has to be solved for the unknown n element vector $\boldsymbol{x}$, where $\boldsymbol{L}$ is an $n \times n$ non-singular lower triangular matrix, then the first element of the unknown vector, x_1 can be calculated from multiplication of the first row of $\boldsymbol{L}$ with the vector $\boldsymbol{x}$:

$$l_{11}x_1 = b_1 \quad \Rightarrow \quad x_1 = \frac{b_1}{l_{11}} \tag{285}$$

The second element can therefore be calculated from multiplication of the second row of $\boldsymbol{L}$ with vector $\boldsymbol{x}$, and substitution of x_1 from Eq. 285:

$$l_{21}x_1 + l_{22}x_2 = b_2 \quad \Rightarrow \quad x_2 = \frac{b_2 - l_{21}\left(\frac{b_1}{l_{11}}\right)}{l_{22}} \tag{286}$$

In general it can be shown that all elements of $\boldsymbol{x}$ can be calculated sequentially from:

$$x_i = \frac{b_i - \sum_{j=1} l_{ij}x_j}{l_{ii}} \tag{287}$$

This method of solving an upper triangular system of linear equations is called forward substitution. Note that if the diagonal elements of $\boldsymbol{L}$ are very small relative to the off-diagonal elements, then the arithmetic required for the computation may require a large dynamic range. See also Matrix Decompositions - Back-Substitution/Cholesky/Gaussian Elimination/QR.

Matrix Decompositions: There are a number of methods which allow a matrix to be decomposed into structured matrices. The reason for performing a *matrix decomposition* is to either extract certain parameters from the matrix, or to provide a computationally cost effective and, ideally,

numerically stable method of solving a set of linear equations. A number of decompositions often performed in DSP can be identified.

- **Back Substitution:** See *Matrix Algorithms - Backsubstitution*.
- **Cholesky:** The *Cholesky* decomposition or factorization can be applied to a $n \times n$ non-singular symmetric (and therefore positive definite) matrix, $\boldsymbol{A}$ such that:

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T = \begin{bmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} & \dots & l_{n1} \\ 0 & l_{22} & l_{32} & \dots & l_{n2} \\ 0 & 0 & l_{33} & \dots & l_{n3} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & l_{nn} \end{bmatrix} \tag{288}$$

If a system of equations, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is to be solved for the unknown n element vector $\boldsymbol{x}$, where $\boldsymbol{A}$ is an $n \times n$ symmetric matrix, and $\boldsymbol{b}$ a known n element vector, the solution can be found by Cholesky factoring matrix $\boldsymbol{A}$, and performing a backsubtitution followed by forward substitution:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \quad \Rightarrow \quad \boldsymbol{L}\boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{b} \quad \Rightarrow \quad \begin{cases} \boldsymbol{L}\boldsymbol{y} = \boldsymbol{b} & \text{solve by forward substitution} \\ \boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{y} & \text{solve by backward substitution} \end{cases} \tag{289}$$

The elements of the Cholesky matrix, $\boldsymbol{L}$, are well bounded and in general Cholesky factorization is a numerically well behaved algorithm with fixed point arithmetic.

The Cholesky factorization may also be written in the form of the LDL^T factorization, where $\boldsymbol{L}$ is now a unit upper triangular matrix, and $\boldsymbol{D}$ is a diagonal matrix. See also Matrix Decompositions - Back Substitution/ Forward Substitution/Gaussian Elimination/LDU/LU/LDLT, Recursive Least Squares - Square Root Covariance.

- **Complete Pivoting:** See entry for Matrix Decompositions - Pivoting.
- **Eigenanalysis:** Eigenanalysis allows a square $n \times n$ matrix, $\boldsymbol{A}$, to be broken down into components of an eigenvector and an eigenvalue which satisfy the condition:

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x} \tag{290}$$

where $\boldsymbol{x}$ is an $n \times 1$ vector, referred to as a (right) eigenvector of $\boldsymbol{A}$, and the scalar λ is an eigenvalue of $\boldsymbol{A}$. In order to calculate the eigenvalues Eq. 290 can be rearranged to give:

$$(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{x} = \boldsymbol{0} \tag{291}$$

and if $\boldsymbol{x}$ is to be a non-zero vector, then the solution to Eq. 291 requires that the matrix $(\boldsymbol{A} - \lambda\boldsymbol{I})$ is singular (i.e. linearly dependent columns) and therefore the determinant is zero, i.e.

$$\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0 \tag{292}$$

This equation is often referred to as the characteristic equation of the matrix $\boldsymbol{A}$, and can be expressed as a polynomial of order n, which in general has n distinct roots. (If the eigenvalue does not have n distinct roots, then the matrix $\boldsymbol{A}$ is said to be degenerate). Therefore we can note that there are n instances of Eq. 290:

$$\boldsymbol{A}\boldsymbol{x}_i = \lambda_i\boldsymbol{x}_i \quad \text{for } i = 1 \text{ to } n \tag{293}$$

Writing the eigenvalues as a diagonal matrix, $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, \dots\lambda_n)$, and each vector, $\boldsymbol{x}_i$ as a column of an $n \times n$ matrix $\boldsymbol{X}$:

$$\boldsymbol{A}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots, \boldsymbol{x}_n) = \boldsymbol{A}\boldsymbol{X} = \boldsymbol{X}\Lambda \tag{294}$$

and therefore $\boldsymbol{X}$ is a similarity transform matrix:

$$\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} = \Lambda \tag{295}$$

and matrices $\boldsymbol{A}$ and Λ are said to be similar. Note also that

$$\text{trace}(\boldsymbol{A}) = \text{trace}(\Lambda) = \lambda_1 + \lambda_2 + \ldots + \lambda_n, \tag{296}$$

which is easily seen from noting that:

$$\text{trace}(\Lambda) = \text{trace}(\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}) = \text{trace}(\boldsymbol{A}\boldsymbol{X}^{-1}\boldsymbol{X}) = \text{trace}(\boldsymbol{A}) \tag{297}$$

For the general eigenvalue problem, techniques such as the QL algorithm (not to be confused with the QR decomposition) are used to reduce the matrix $\boldsymbol{A}$ to various structured intermediate forms before ultimately extracting eigenvalues and eigenvectors. Note that although the eigenvalues could be found from solving the polynomial in Eq. 292 this is in general not a good method either numerically or computationally.

For DSP systems a particularly relevant problem is the symmetric eigenvalue problem, whereby a (symmetric) correlation matrix is to be decomposed. For a symmetric $n \times n$ matrix $\boldsymbol{R}$,

$$\boldsymbol{R}\boldsymbol{q}_i = \lambda \boldsymbol{q}_i \quad \text{for } i = 1 \text{ to } n \tag{298}$$

it is relatively straightforward to show for the symmetric case that the eigenvectors, $\boldsymbol{q}_i$, will be orthogonal to each other, and Eq. 295 can be written in the form:

$$\boldsymbol{Q}^T\boldsymbol{R}\boldsymbol{Q} = \Lambda \quad \text{or} \quad \boldsymbol{R} = \boldsymbol{Q}\Lambda\boldsymbol{Q}^T \tag{299}$$

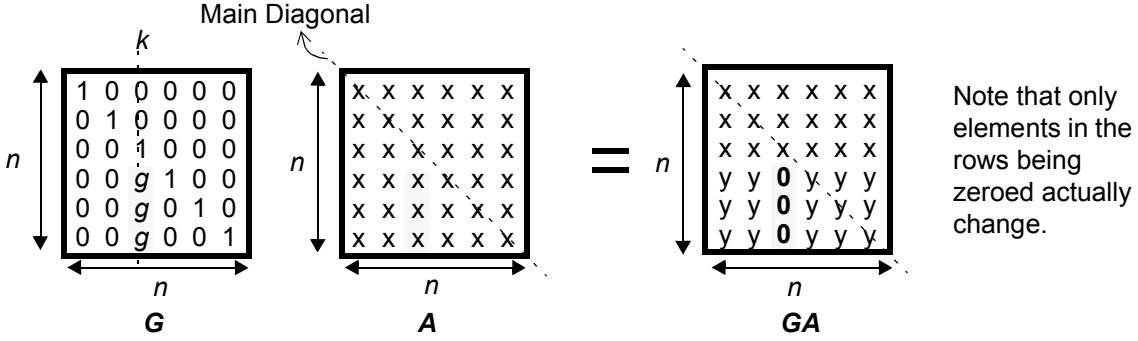
where, $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$.

Other useful properties of the symmetric eigenanalysis problems are that the condition number of $\boldsymbol{R}$ can be calculated as the eigenvalue spread:

$$\kappa(\boldsymbol{R}) = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{300}$$

See also Matrix Decompositions - Singular Value, QL, QR Algorithm.

- **Schur Form:** A canonical form of a matrix that displays the eigenvalues but not eigenvectors of matrix.
- **Eigenvalue:** See Matrix Decompositions - Eigenanalysis.
- **Eigenvector:** See Matrix Decompositions - Eigenanalysis.
- **Fast Given's Rotations:** See Matrix Decompositions - Square Root Free Givens.
- **Forward Substitution:** See Matrix Algorithms - Forward Substitution.
- **Gauss Transform:** In general the *Gauss transform*, G_k is an $n \times n$ matrix used to zero the $k-1$ elements below the main diagonal in column k of a non-singular $n \times n$ matrix, $\boldsymbol{A}$:



As an example of zeroing a matrix column below the main diagonal, the elements a_{31} and a_{21} can be "zeroed" by premultiplying a 3×3 matrix $\boldsymbol{A}$ with a 3×3 *Gauss transform* matrix, $\boldsymbol{G}_1$:

$$\begin{aligned} \boldsymbol{G}_1\boldsymbol{A} &= \begin{bmatrix} 1 & 0 & 0 \\ g_{21} & 1 & 0 \\ g_{31} & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ (a_{11}g_{21} + a_{21}) & (a_{12}g_{21} + a_{22}) & (a_{13}g_{21} + a_{23}) \\ (a_{11}g_{31} + a_{31}) & (a_{12}g_{31} + a_{32}) & (a_{13}g_{31} + a_{33}) \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & b_{32} & b_{32} \end{bmatrix} \quad \text{where,} \quad g_{31} = -\frac{a_{31}}{a_{11}} \quad \text{and} \quad g_{21} = -\frac{a_{21}}{a_{11}} \end{aligned} \tag{301}$$

In general the *Gauss transform* matrix which will zero all elements below the diagonal in the *k*-th column of an $n \times n$ matrix, $\boldsymbol{A}$ can be specified as:

$$\begin{aligned} \boldsymbol{G}_k\boldsymbol{A} &= \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & g_{k+1,k} & 1 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & g_{nk} & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{k1} & \cdots & a_{kk} & a_{k,k+1} & \cdots & a_{kn} \\ a_{k+1,1} & \cdots & a_{k+1,k} & a_{k+1,k+1} & \cdots & a_{k+1,n} \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nk} & a_{n,k+1} & \cdots & a_{nn} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{k1} & \cdots & a_{kk} & a_{k,k+1} & \cdots & a_{kn} \\ b_{k+1,1} & \cdots & 0 & b_{k+1,k+1} & \cdots & b_{k+1,n} \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ b_{n1} & \cdots & 0 & b_{n,k+1} & \cdots & b_{nn} \end{bmatrix} \end{aligned} \tag{302}$$

where $g_{ik} = -a_{ik} / a_{kk}$.

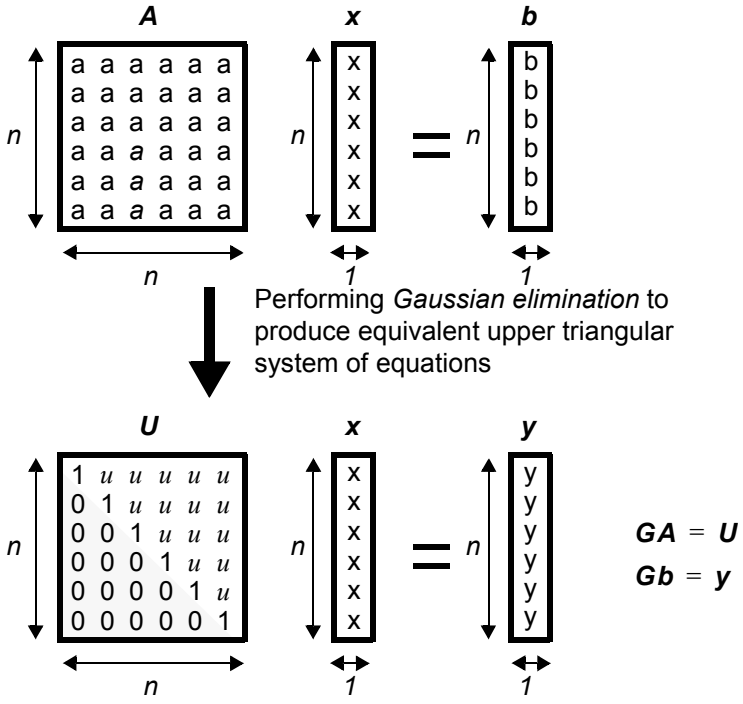
The inverse of a *Gauss transform* matrix, $\boldsymbol{G}_k^{-1}$ is simply calculated by negating the "g"entries:

$$\boldsymbol{G}_k^{-1} = -\begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -g_{k+1,k} & 1 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & -g_{nk} & 0 & \cdots & 1 \end{bmatrix} \tag{303}$$

Gauss transforms are used in the main for performing LU matrix decomposition. *Gauss transforms* are not in general numerically well behaved, and if the pivot element (the divisor a_{ii}) is very small in magnitude, then very large values may occur in the resulting transformed matrix; hence "pivoting" strategies are often used whereby rows and/or columns of the matrix are interchanged, but the integrity of the problem being solved is maintained. See also Matrix Decompositions - Gaussian Elimination/LU/Pivotting, Matrix Structured - Lower Triangular/Upper Triangular.

- **Gaussian Elimination:** *Gaussian elimination* is a technique used to find the solution of a square set of linear equations, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, for the unknown *n* element vector $\boldsymbol{x}$, where $\boldsymbol{A}$ is an $n \times n$ non-singular matrix, and $\boldsymbol{b}$ a known *n* element vector. *Gaussian elimination* converts a square non-singular matrix into an equivalent, and easier to solve system of equations where $\boldsymbol{A}$ has been implicitly premultiplied by a matrix,

G, to produce an upper triangular matrix, ***U*** and a new vector ***y***. (Note that the premultiplication is described as "implicit" as it is not necessary to explicitly form the matrix ***G*** - the *Gaussian elimination* is done in stages.).



Gaussian elimination can be formally described in terms of Gauss transforms which are used to "zero" the elements below the main diagonal of a matrix to ultimately convert it to an upper triangular form using a series of Gauss transforms for each column of the matrix.

The Gauss transform matrix, $\boldsymbol{G}_k$ can be specified which will zero all elements below the diagonal in the k-th. column of an $n \times n$ matrix, ***A***. Therefore to solve the system of linear equations, $\boldsymbol{Ax} = \boldsymbol{b}$, the transforms $\boldsymbol{G}_1$ to $\boldsymbol{G}_k$ can be used to premultiply matrix ***A*** (in the correct order) such that:

$$\begin{aligned} \boldsymbol{Ax} &= \boldsymbol{b} \\ \Rightarrow \boldsymbol{G}_{n-1} \ldots \boldsymbol{G}_2 \boldsymbol{G}_1 \boldsymbol{Ax} &= \boldsymbol{G}_{n-1} \ldots \boldsymbol{G}_2 \boldsymbol{G}_1 \boldsymbol{b} \\ \Rightarrow \boldsymbol{Ux} &= \boldsymbol{y} \end{aligned} \tag{304}$$

and the equivalent system of equations, $\boldsymbol{Ux} = \boldsymbol{y}$ is solved by backsubstitution.

In general *Gaussian elimination* is not numerically well behaved, and will fail if ***A*** is singular. In particular small pivot elements, a_{ii} on the diagonal of matrix ***A*** may lead to very small and very large values appearing in the ***L*** and ***U*** matrices respectively. Therefore pivoting techniques are often used whereby the rows and/or columns of ***A*** are interchanged using (orthogonal) permutation matrices. In fact where *Gaussian elimination* is to be used for solving a set of linear equations, it is recommended that pivoting is always used. See also Matrix Decompositions - Gauss Transforms/LU/Pivotting, Matrix Structured - Lower Triangular/Upper Triangular.

- **Givens Rotations:** *Given's rotations* (also known as plane rotations, and Jacobi rotations) represent an orthogonal transformation for introducing zero elements into a matrix. The element a_{21} of the following (full rank) matrix can be zeroed by applying the appropriate *Givens rotation* as follows:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}\begin{bmatrix} a_{11} & a_{12} \ldots a_{1n} \\ a_{21} & a_{22} \ldots a_{2n} \end{bmatrix}$$

$$= \begin{bmatrix} \sqrt{a_{11}^2 + a_{21}^2} & (a_{12}c + sa_{22}) & \ldots & (a_{1n}c + sa_{2n}) \\ 0 & (-sa_{12} + ca_{22}) & \ldots & (-sa_{1n} + ca_{2n}) \end{bmatrix} \tag{305}$$

$$= \begin{bmatrix} b_{11} & b_{12} \ldots b_{1n} \\ 0 & b_{22} \ldots b_{2n} \end{bmatrix}$$

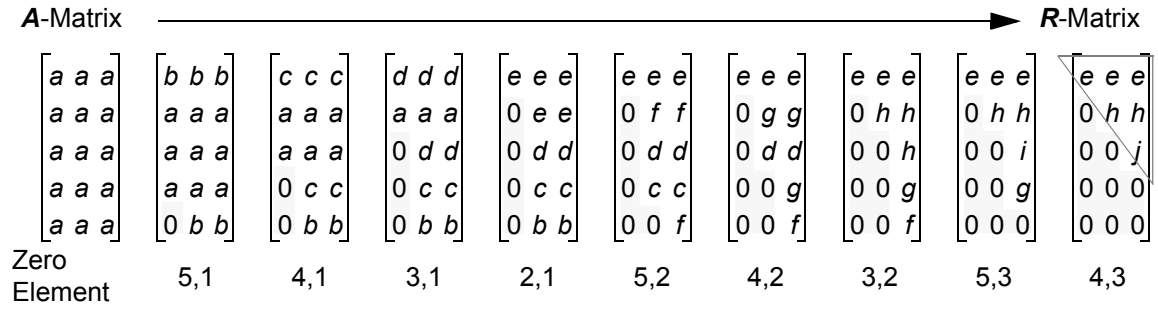
where

$$c = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}} \quad \text{and} \quad s = \frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}} \tag{306}$$

More generally if a zero is to be introduced in the *i*-th row and *j*-th column of an $m \times n$ matrix ***A*** by rotating with the element in the *k*-th row and *j*-th column, then an $m \times m$ *Given's rotation* matrix, ***G***, can be applied:

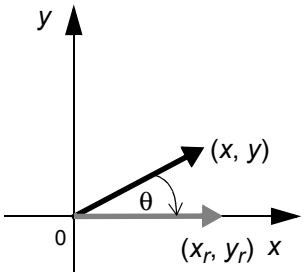
$$\boldsymbol{GA} = \begin{array}{c} \\ k \\ \\ i \\ \\ m \end{array}\begin{array}{c} \begin{array}{ccccccc} k & & i & & m \end{array} \\ \begin{bmatrix} 1 & \ldots & 0 & \ldots & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & c & \ldots & s & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & -s & \ldots & c & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & \ldots & 0 & \ldots & 1 \end{bmatrix} \end{array}\begin{bmatrix} a_{11} & \ldots & a_{1j} & \ldots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ a_{k1} & \ldots & a_{kj} & \ldots & a_{kn} \\ \vdots & & \vdots & & \vdots \\ a_{i1} & \ldots & a_{ij} & \ldots & a_{in} \\ \vdots & & \vdots & & \vdots \\ a_{m1} & \ldots & a_{mj} & \ldots & a_{mn} \end{bmatrix} \quad \text{where, } c = \frac{a_{kj}}{\sqrt{a_{kj}^2 + a_{ij}^2}} \text{ and } s = \frac{a_{ij}}{\sqrt{a_{kj}^2 + a_{ij}^2}} \tag{307}$$

Given's rotations are particularly useful for realizing the upper triangular ***R*** matrix in a QR decomposition algorithm. Consider that a 5×3 full rank matrix is to be decomposed into its ***Q*** and ***R*** components (for notational clarity all matrix variables row-column subscripts have been omitted):



All of the elements below the main diagonal in column 1 are first rotated with the a_{11} element and after four Given's rotations all appropriate elements are zeroed. For column 2, all elements below the main diagonal are rotated with the e_{22} element, and after three *Given's rotations* all appropriate elements are zeroed. Finally for column 3, all elements below the main diagonal are rotated with i_{33} and after two *Given's rotations* the upper triangular matrix ***R*** is realized. Note that the order of element rotation is important in order that previously zeroed elements are retained as zeroes when subsequent columns are rotated. Also note that when a matrix is rotated the only elements that change are the ones in the row with the element being zeroed, and the row with which the element is being rotated. Finally if the ***Q*** matrix is specifically required, then the *Given's rotation* (sparse) matrices of the form in Eq. 307 can be retained and multiplied together at a later stage.

The name *Given's rotations* is after W. Givens , and the word rotation is used because the transform corresponds to an angle rotation of a vector $[x, y]^T$ in the x-y plane to the vector $[x_r, y_r]^T$ by an angle of θ; this also explains the name "plane rotation".



$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

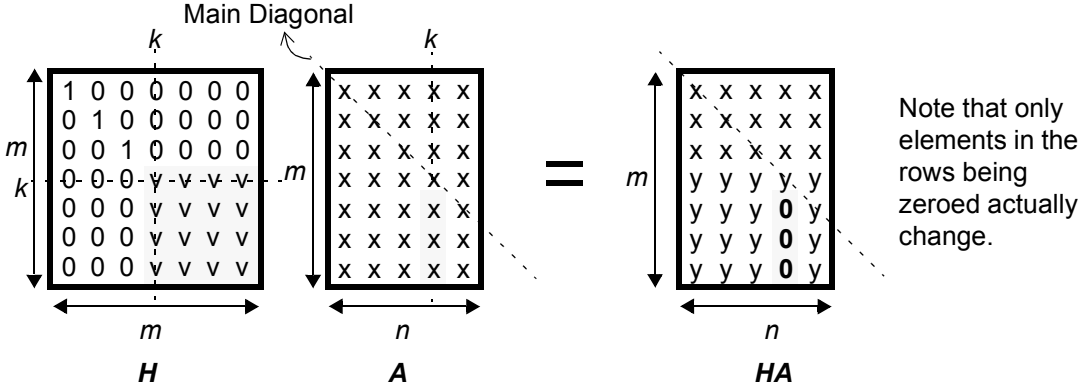
$$\cos\theta = \frac{x}{\sqrt{x^2+y^2}} \qquad \sin\theta = \frac{y}{\sqrt{x^2+y^2}}$$

$$\theta = \tan^{-1}\frac{y}{x}$$

Because of the orthogonal nature of the Given's rotations, the technique is numerically well behaved. From an intuitive consideration of Eq. it can be seen that the magnitude of *c* and *s* will always be less than one (i.e. $|c| < 1$ and $|s| < 1$) and therefore elements in the transformed matrix will have adequately bounded values.

Over the last few years *Given's rotations* have been widely used for adaptive signal processing problems where fast numerically stable parallel algorithms have been required. See also Matrix Decompositions - QR, Recursive Least Squares - QR.

- **Householder Transformation:** The *Householder transformation* is an $m \times m$ matrix, $\boldsymbol{H}$, used to zero the elements below the main diagonal in the *k*-th row of a full rank $m \times n$ matrix $\boldsymbol{A}$:



Householder matrices are orthogonal, i.e. $\boldsymbol{H}\boldsymbol{H}^T = \boldsymbol{I}$, and also symmetric, i.e. $\boldsymbol{H} = \boldsymbol{H}^T$. The *Householder transformation* can be illustrated by noting that the $k-1$ lower elements of a $k \times 1$ vector, $\boldsymbol{x}$, can be zeroed by premultiplying with a suitable Householder matrix:

$$\boldsymbol{H}\boldsymbol{x} = \left(\boldsymbol{I} - \frac{2\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}}\right)\boldsymbol{x} = \left(\boldsymbol{I} - \frac{2\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}}\right)\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{x}\|_2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{308}$$

where

$$\boldsymbol{v} = \begin{bmatrix} x_1 + \|\boldsymbol{x}\|_2 \\ x_2 \\ x_3 \\ \vdots \\ x_k \end{bmatrix} \tag{309}$$

and the 2-norm, $\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2 + x_3^2 + \ldots + x_k^2}$.

Therefore the general Householder matrix, $\boldsymbol{H}_k$, to zero the elements in column k, below the main diagonal of a matrix $\boldsymbol{A}$ can be written in a partitioned matrix form:

$$\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_{kk} \end{bmatrix} \left[\begin{array}{ccc|ccc} a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2k} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \hline a_{k1} & a_{k2} & \cdots & a_{kk} & \cdots & a_{kn} \\ a_{k+1,1} & a_{k+1,2} & \cdots & a_{k+1,k} & \cdots & a_{k+1,n} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} & \cdots & a_{m,n} \end{array}\right] = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_{kk} \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{1k} \\ \boldsymbol{A}_{k1} & \boldsymbol{A}_{kk} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{1k} \\ \boldsymbol{H}_{kk}\boldsymbol{A}_{k1} & \boldsymbol{H}_{kk}\boldsymbol{A}_{kk} \end{bmatrix}$$

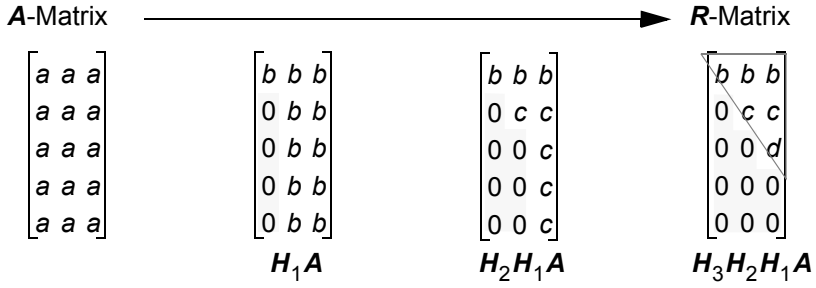
$$= \left[\begin{array}{ccc|ccc} a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2k} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \hline b_{k1} & b_{k2} & \cdots & b_{kk} & \cdots & b_{kn} \\ b_{k+1,1} & b_{k+1,2} & \cdots & 0 & \cdots & b_{k+1,n} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ b_{m1} & b_{m2} & \cdots & 0 & \cdots & b_{m,n} \end{array}\right] \tag{310}$$

where,

$$\boldsymbol{H}_{kk} = \boldsymbol{I} - \frac{2\boldsymbol{v}_k\boldsymbol{v}_k^T}{\boldsymbol{v}_k^T\boldsymbol{v}_k} \quad \text{with} \quad \boldsymbol{v}_k = \begin{bmatrix} a_{kk} + \|\boldsymbol{a}_{kk}\|_2 \\ a_{k+1,k} \\ \vdots \\ a_{mk} \end{bmatrix} \quad \text{and} \quad \boldsymbol{a}_{kk} = \begin{bmatrix} a_{kk} \\ a_{k+1,k} \\ \vdots \\ a_{mk} \end{bmatrix} \tag{311}$$

where $\|\boldsymbol{a}_{kk}\|_2 = \sqrt{a_{kk}^2 + a_{k,k+1}^2 + \ldots + a_{mk}^2}$.

A sequence of Householder matrices is very useful for performing certain matrix transforms such as the QR decomposition. Consider an example where a 5×3 full rank matrix is to be decomposed into its $\boldsymbol{Q}$ and $\boldsymbol{R}$ components (for clarity all matrix variable row-column subscripts have been omitted):



Compared to Given's rotations, which zero a column vector element by element, *Householder transformations* requires fewer arithmetic operations, however Given's rotations have become more popular for modern DSP techniques as a result of their suitability for parallel array implementation [77], [88], unlike the Householder transformation which has no recursive implementation.

The zeroing of column elements in a matrix can also be performed by the Gauss transform typically for implementation of algorithms such as LU decomposition. However unlike the *Householder transform*, Gauss transforms are not orthogonal. Therefore because the *Householder transform* does not produce matrices with very large or very small elements (which may happen with the Gauss transform) then the numerical behavior is in general good [136]. See also Matrix Decompositions - Given's Rotations/QR/ SVD/, Recursive Least Squares - QR

- **LDL^T:** See Matrix Decompositions - Cholesky.
- **LDU:** LDU decomposition is a special case of LU decomposition, whereby a non-singular matrix $n \times n$ $\boldsymbol{A}$, can be factored into a unit upper triangular matrix $\boldsymbol{L}$, a unit lower triangular matrix $\boldsymbol{U}$, and a diagonal matrix, $\boldsymbol{D}$. See also Matrix Decompositions - Cholesky/LU.
- **LL^T:** See Matrix Decompositions - Cholesky.
- **LU:** The LU decomposition is used to convert a non-singular $n \times n$ matrix $\boldsymbol{A}$, into a lower and upper triangular matrix product:

$$\boldsymbol{A} = \boldsymbol{LU} = \begin{bmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix} \tag{312}$$

Gaussian elimination (or factorization), via a series of Gauss transforms, can be used to produce the LU decomposition. The k-th Gauss transform matrix, $\boldsymbol{G}_k$ will zero all of the elements below the main diagonal in the k-th column of an $n \times n$ matrix, $\boldsymbol{A}$. After applying the Gauss transforms $\boldsymbol{G}_1$ to $\boldsymbol{G}_{n-1}$ an upper triangular matrix is produced:

$$\boldsymbol{G}_{n-1} \dots \boldsymbol{G}_2 \boldsymbol{G}_1 \boldsymbol{A} = \boldsymbol{U} \tag{313}$$

To obtain the lower triangular matrix, the above equation can be rearranged to go:

$$\begin{aligned} \Rightarrow \boldsymbol{A} &= \boldsymbol{G}_1^{-1} \boldsymbol{G}_2^{-1} \dots \boldsymbol{G}_{n-1}^{-1} \boldsymbol{U} \\ \Rightarrow \boldsymbol{A} &= \boldsymbol{LU} \end{aligned} \tag{314}$$

where $\boldsymbol{L} = \boldsymbol{G}_1^{-1} \boldsymbol{G}_2^{-1} \dots \boldsymbol{G}_{n-1}^{-1}$.

Note that the inverse Gauss transform matrices, $\boldsymbol{G}_i^{-1}$ are trivial to compute from $\boldsymbol{G}_i$. and they will also be lower triangular matrices (the product of two lower triangular matrices is always lower triangular).

If a system of equations, $\boldsymbol{Ax} = \boldsymbol{b}$ is to be solved for the unknown n element vector $\boldsymbol{x}$, where $\boldsymbol{A}$ is an $n \times n$ non-singular matrix, and $\boldsymbol{b}$ a known n element vector, the solution can be found by LU factoring matrix $\boldsymbol{A}$, and performing a backsubtitution and a forward substitution:

$$\boldsymbol{Ax} = \boldsymbol{b} \quad \Rightarrow \quad \boldsymbol{LUx} = \boldsymbol{b} \quad \Rightarrow \quad \begin{cases} \boldsymbol{Ly} = \boldsymbol{b} & \text{solve by forward substitution} \\ \boldsymbol{Ux} = \boldsymbol{y} & \text{solve by backward substitution} \end{cases} \tag{315}$$

It is however less computation to perform Gaussian elimination which form the U matrix, but does not explicitly form the $\boldsymbol{L}$ matrix. In general using LU decomposition (or Gaussian elimination) to solve a system of linear equations does not have good numerical behavior and the existence of small elements on the diagonals of $\boldsymbol{L}$ and $\boldsymbol{U}$, and large values elsewhere may lead to the computation requiring a very large

dynamic range. Therefore pivoting techniques are usually used on the Gaussian elimination computation in an attempt to circumvent the effects of small and large values.

See also Matrix Decompositions - Backsubstitution/Cholesky/Forward Substitution/Gaussian Elimination/ LDU/LDLT/Pivoting.

- **Partial Pivoting:** See entry for Matrix Decompositions - Pivoting.
- **Pivoting:** When performing certain forms of matrix decomposition such as LU, small elements on the main diagonal are used as divisors when producing matrices such as Gauss transforms to zero certain elements in the matrix. If these elements are very small then they can result in very large numbers appearing in the matrices resulting from the decomposition.

  For example consider the LU decomposition of the following 3×3 matrix:

$$\boldsymbol{A} = \begin{bmatrix} 0.0001 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 10000 & 1 & 0 \\ 10000 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.0001 & 1 & 1 \\ 0 & -9999 & -9998 \\ 0 & 0 & 1 \end{bmatrix} = \boldsymbol{LU} \tag{316}$$

  If fixed point arithmetic is used, then the dynamic range of numbers required for the $\boldsymbol{L}$ and $\boldsymbol{U}$ matrices is twice that for the $\boldsymbol{A}$ matrix. Small pivot elements can be avoided by rearranging the $\boldsymbol{A}$ matrix elements using orthogonal permutation matrices. Therefore for the above example:

$$\boldsymbol{PA} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.0001 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 3 \\ 0.0001 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.0001 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 3 \\ 0 & 0.9999 & 0.9997 \\ 0 & 0 & -1 \end{bmatrix} = \boldsymbol{L}_p\boldsymbol{U}_p \tag{317}$$

  and the LU factors now contain suitably small elements. In general when performing pivoting, prior to applying the Gauss transform on the k-th column, the column is scanned to find the smallest element in order to set up the permutation matrix to appropriately swap the rows and attempt to ensure that small pivots are avoided. If a system of linear equations:

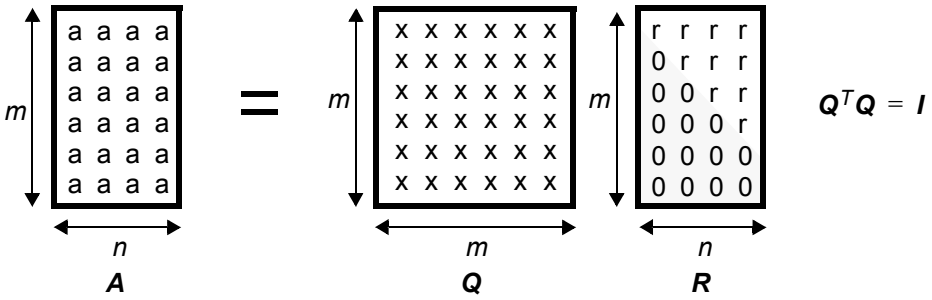
$$\boldsymbol{Ax} = \boldsymbol{b} \tag{318}$$

  is to be solved using Gaussian elimination (or more exactly LU decomposition with one stage of pivoting), where $\boldsymbol{A}$ is a non-singular $n \times n$ matrix, $\boldsymbol{b}$ is a known n element vector, and $\boldsymbol{x}$ is an unknown n element vector then:

$$\boldsymbol{PAx} = \boldsymbol{Pb} \quad \Rightarrow \quad \boldsymbol{LUx} = \boldsymbol{Pb} \quad \Rightarrow \quad \begin{cases} \boldsymbol{Ly} = \boldsymbol{Pb} & \text{solve by forward substitution} \\ \boldsymbol{Ux} = \boldsymbol{y} & \text{solve by backward substitution} \end{cases} \tag{319}$$

  If both the rows and the columns are scanned to circumvent small pivots, then this is often referred to as *complete pivoting*. Column swapping is achieved by postmultiplication of matrix $\boldsymbol{A}$, with a suitable permutation matrix $\boldsymbol{Q}$. Pivoting can be used on many other linear algebraic decompositions where small pivoting/divisor elements need to be avoided. Note that because the pivot matrix $\boldsymbol{P}$ (and also $\boldsymbol{Q}$) is orthogonal, then for least squares type operations, the 2-norm of the pivoted matrix, $\boldsymbol{PA}$ is not affected. See also Matrix Decomposition - Gaussian Elimination/LU, Vector Properties - Norm.
- **Plane Rotations:** See entry for Matrix Decompositions - Given's Rotations.

- **QR:** The QR matrix decomposition is an extremely useful technique in least squares signal processing systems where a full rank $m \times n$ matrix $\boldsymbol{A}$ $(m > n)$ is decomposed into an upper triangular matrix, $\boldsymbol{R}$ and an orthogonal matrix $\boldsymbol{Q}$:



If the least squares solution is required for the overdetermined linear set of equations:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{320}$$

where $\boldsymbol{A}$ is an $m \times n$ matrix, $\boldsymbol{b}$ is a known m element vector, and $\boldsymbol{x}$ is an unknown n element vector, then the minimum norm solution is required, i.e. minimize, ε, where $\varepsilon = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$. This can be found by the least squares solution:

$$\boldsymbol{x}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{321}$$

However noting that the 2-norm (or Euclidean norm) is invariant under orthogonal transforms, then the QR decomposition allows a different computation method to find the solution. Using a suitable sequence of Given's rotations, or Householder rotations, for a full rank $m \times n$ matrix $\boldsymbol{A}$ (where $m > n$), the QR decomposition yields:

$$\boldsymbol{A} = \boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{bmatrix}, \quad \text{or} \tag{322}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \\ a_{n+1,1} & a_{n+1,2} & \cdots & a_{n+1,n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} & q_{1,n+1} & \cdots & q_{1m} \\ q_{21} & q_{22} & \cdots & q_{2n} & q_{2,n+1} & \cdots & q_{2m} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ q_{n1} & q_{n2} & \cdots & q_{nn} & q_{n,n+1} & \cdots & q_{nm} \\ q_{n+1,1} & q_{n+1,2} & \cdots & q_{n+1,n} & q_{n+1,n+1} & \cdots & q_{n+1,m} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ q_{m1} & q_{m2} & \cdots & q_{mn} & q_{m,n+1} & \cdots & q_{mm} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & r_{nn} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{323}$$

where $\boldsymbol{Q}$ is an $m \times m$ orthogonal matrix, (i.e. $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}$), and $\boldsymbol{R}$ is an $n \times n$ upper triangular matrix, and $\boldsymbol{0}$ a $(m-n) \times n$ zero matrix, then:

$$\varepsilon = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \|\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{Q}^T\boldsymbol{b}\|_2 = \left\|\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{x} - \begin{bmatrix}\boldsymbol{c}\\ \boldsymbol{d}\end{bmatrix}\right\|_2 = \|\boldsymbol{v}\|_2 \tag{324}$$

where $\boldsymbol{c}$ is an n element vector, and $\boldsymbol{d}$ and $m-n$ element vector and vector $\boldsymbol{v}$ is therefore computed as:

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \\ v_{n+1} \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & r_{nn} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ d_1 \\ \vdots \\ d_{m-n} \end{bmatrix} = \begin{bmatrix} r_{11}x_1 + r_{12}x_2 + \dots + r_n x_n \\ r_{22}x_2 + \dots + r_{2n}x_n \\ \vdots \\ r_{nn}x_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ d_1 \\ \vdots \\ d_{m-n} \end{bmatrix} \tag{325}$$

In order to minimize $\|\boldsymbol{v}\|_2$, note that:

$$\|\boldsymbol{v}\|_2^2 = \|\boldsymbol{R}\boldsymbol{x} - \boldsymbol{c}\|_2^2 + \|\boldsymbol{d}\|_2^2 \tag{326}$$

Therefore solving the system of equations $\boldsymbol{R}\boldsymbol{x} - \boldsymbol{c} = \boldsymbol{0}$ will give the desired least squares solution of $\|\boldsymbol{v}\|_2$ (note that the sub-vector norm $\|\boldsymbol{d}\|_2$ cannot be minimized) i.e.,

$$\boldsymbol{x}_{LS} = \boldsymbol{R}^{-1}\boldsymbol{c} \tag{327}$$

which can be conveniently solved using backsubstitution rather than performing the explicit inverse. The least squares residual is simply the value $\|\boldsymbol{d}\|_2$.

Because of the orthogonal nature of the algorithm, the QR is numerically well behaved and represents an extremely powerful and versatile basis for least squares signal processing techniques. Also a brief comparison of the solution obtained in Eq. 321 and that of Eqs. 322-327 will show that the *QR* approach operates directly on the data matrix, whereas the pseudoinverse form in Eq. 321 requires to square the matrix $\boldsymbol{A}$. Therefore a simplistic argument is that twice the dynamic range is required to accommodate the spread of numerical values in the pseudoinverse method, as compared to the *QR* based least squares solution. (Note that both solutions are identical if infinite precision arithmetic is used.)

See also Least Squares, Matrix Decompositions - Back substitution/Given's Rotation/Pseudoinverse, Matrix Properties - Overdetermined, Recursive Least Squares - QR.

- **Similarity Transform:** Two non-singular $n \times n$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are said to similar is there exists a similarity transform matrix $\boldsymbol{X}$, such that:

$$\boldsymbol{B} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X} \tag{328}$$

See also Matrix Decompositions - Eigenanalysis.

- **Singular Value:** The *singular value decomposition* (*SVD)* is one of the most important and useful decompositions in linear algebraic theory. The SVD allows an $m \times n$ matrix $\boldsymbol{A}$, with $r = \text{rank}(\boldsymbol{A}) \le \min(m, n)$ to be transformed in the following manner:

$$\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V} = \begin{bmatrix} \Sigma & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{329}$$

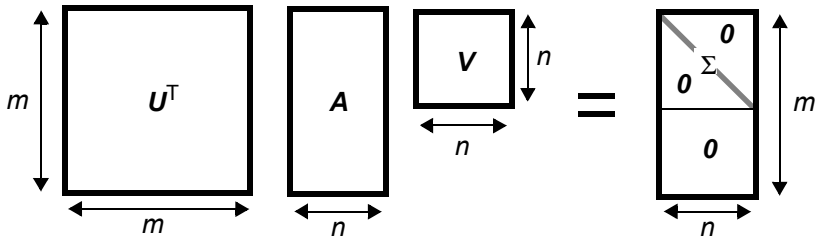
and therefore:

$$\boldsymbol{A} = \boldsymbol{U}\begin{bmatrix} \Sigma & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\boldsymbol{V}^T \tag{330}$$

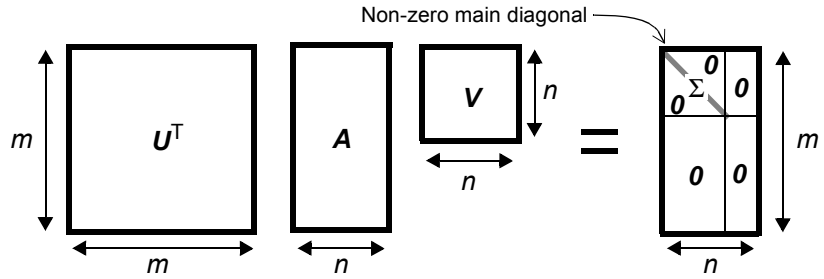
where $\boldsymbol{U}$ is an $m \times m$ orthogonal matrix, i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, $\boldsymbol{V}$ is an $n \times n$ orthogonal matrix, i.e. $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$, and Σ is a diagonal sub-matrix containing the singular values of $\boldsymbol{A}$:

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_r) \tag{331}$$

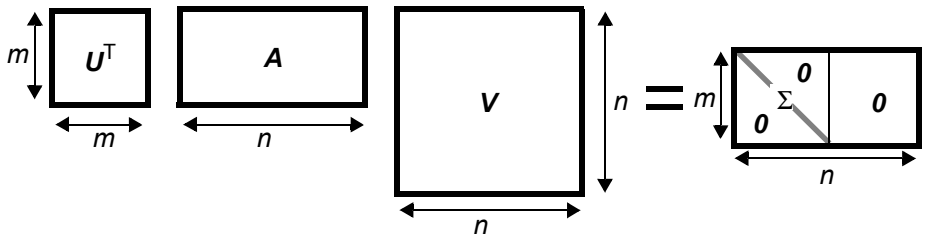
The Σ matrix is usually written such that $\sigma_1 > \sigma_2 > \ldots > \sigma_r$. The *singular value decomposition* can be illustrated in a more diagrammatic form. If for matrix $\boldsymbol{A}$, $m > n$, and $r = \text{rank}(\boldsymbol{A}) = n$ the Σ matrix has all non-zero elements in the main diagonal:



Note if $r < n$ then $\boldsymbol{A}$ has linearly dependent columns and there will be only r non-zero elements:



If for matrix $\boldsymbol{A}$, $m < n$, and $r = \text{rank}(\boldsymbol{A}) = m$ the Σ matrix has all non-zero elements in the main diagonal (again, note if $r < m$ then $\boldsymbol{A}$ has linearly dependent columns and there will be only r non-zero elements):



For signal processing algorithms, one of the main uses of the *SVD* is the definition of the pseudoinverse, $\boldsymbol{A}^+$ which can be used to provide the least squares solution to a system of linear equations of the form:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{332}$$

where $\boldsymbol{A}$ is an $m \times n$ matrix, $\boldsymbol{b}$ is a known m element vector, and $\boldsymbol{x}$ is an unknown n element vector. The least squares, minimum norm solution is given by:

$$\boldsymbol{x} = \boldsymbol{A}^+\boldsymbol{b} \tag{333}$$

where

$$\boldsymbol{A}^+ = \boldsymbol{V}\begin{bmatrix}\Sigma^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{U}^T \tag{334}$$

If it is assumed that $\boldsymbol{A}$ has full rank, i.e. $\text{rank}(\boldsymbol{A}) = \min(m, n)$ There are three possible cases for the dimensions of matrix $\boldsymbol{A}$, if:

- $m = n$ (square matrix) then $\boldsymbol{A}^+ = \boldsymbol{A}^{-1}$;
- $m > n$ (the overdetermined problem) then $\boldsymbol{A}^+ = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$, and
- $m < n$ (the underdetermined problem) then $\boldsymbol{A}^+ = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}$.

The transformation of the pseudo-inverse in Eq. 334 into the three forms shown above can be confirmed with straightforward linear algebra. Note that if $\boldsymbol{A}$ is rank deficient then none of the above three cases apply and solution can only be found using the pseudoinverse of Eq. 334. In DSP systems the overdetermined problem (such as found in adaptive DSP) is by far the most common and recognizable "least squares solution". However the pseudoinverse also provides a minimum norm solution for the underdetermined problem when $\boldsymbol{A}$ is rank deficient (e.g., inverse modelling problems such as are found in biomedical imaging and seismic data processing).

Note that if a non-singular square $n \times n$ matrix $\boldsymbol{R}$ is symmetric then the eignenvalue decomposition can then be written as:

$$\boldsymbol{R} = \boldsymbol{Q}^T \Lambda \boldsymbol{Q} \tag{335}$$

where $\Lambda = [\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n]$ and the eigenvalues equal the singular values. If in fact, $\boldsymbol{R} = \boldsymbol{A}^T\boldsymbol{A}$, and $\boldsymbol{A}$ is a full rank $m \times n$ matrix, then the singular values of $\boldsymbol{A}$, are the square roots of the eigenvalues. This can be seen by noting that:

$$\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{V}\begin{bmatrix}\Sigma & \boldsymbol{0}\end{bmatrix}\boldsymbol{U}^T\boldsymbol{U}\begin{bmatrix}\Sigma \\ \boldsymbol{0}\end{bmatrix}\boldsymbol{V}^T = \boldsymbol{V}\begin{bmatrix}\Sigma & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\Sigma \\ \boldsymbol{0}\end{bmatrix}\boldsymbol{V}^T = \boldsymbol{V}\Sigma^2\boldsymbol{V}^T \tag{336}$$

where for illustration purposes $m > n$.

To calculate the singular value decomposition, there are two useful techniques - the Jacobi algorithm and the QR algorithm [15], [77]. See also Least Squares Matrix Properties - Pseudoinverse, Vector Properties - Minimum Norm. .

- **Spectral Decomposition**: The eigenvalue-eigenvector decomposition of a matrix is often referred to as the *spectral decomposition*. See also Matrix Decomposition - Eigenanalysis.
- **Square Root Free Given's Rotations:** *Square root free Given's rotations* (also known as *fast Given's*) are simply a rearranged version of the Given's rotation, where the square root operation has been circumvented, and an additional diagonal matrix introduced [15]. The reason for doing so is that most DSP processors are not optimized for the square root operation, and hence their implementation can be slow. It is worth pointing out that stable versions of the square root free Given's require more divisions per rotation than standard Given's, and DSP processors usually perform square roots faster than divides! Hence the alternative name of *fast Given's*, is not a wholly representative name. It is also worthwhile noting that the *square root free Given's* may have numerical problems of overflow and underflow, unlike the standard Given's rotations. Unless square rooting is impossible, there is probably no good reason to use *square root free Given's rotations*. .
- **Square Root Decomposition**: See entry for Matrix Decompositions - Cholesky Decomposition.
- **Triangularization**: There are a number of matrix decompositions and algorithms which produce factors of a matrix that have upper and lower triangular forms. Any such procedure can therefore be referred to as a Triangularization. See Matrix Decompositions - Cholesky/LU/QR.

Matrix Identities: See *Matrix Properties*.

Matrix Inverse: See *Matrix Properties - Inversion.*

Matrix Inversion Lemma: See *Matrix Properties - Inversion Lemma.*

Matrix Addition: See *Matrix Operations - Addition.*

Matrix Multiplication: See *Matrix Operations - Multiplication.*

Matrix Postmultiplication: See *Matrix Operations - Postmultiplication.*

Matrix Premultiplication: See *Matrix Operations - Premultiplication.*

Matrix Operations: Matrices can be added, subtracted, multiplied, scaled, transposed, and inverted. See also *Matrix Operation Complexity.*

- **Addition (Subtraction):** If two matrices are to be added (or subtracted) then they must be of exactly the same dimensions. Each element in one matrix is added (subtracted) to the analogous element in the other matrix. For example:

$$\begin{bmatrix} 1 & 5 & 4 \\ 6 & 2 & 3 \\ 2 & 8 & 7 \end{bmatrix} + \begin{bmatrix} 3 & 5 & 1 \\ 0 & 2 & 1 \\ 3 & 2 & 0 \end{bmatrix} = \begin{bmatrix} (1+3) & (5+5) & (4+1) \\ (6+0) & (2+2) & (3+1) \\ (2+3) & (8+2) & (7+0) \end{bmatrix} = \begin{bmatrix} 4 & 10 & 5 \\ 6 & 4 & 4 \\ 5 & 10 & 7 \end{bmatrix} \tag{337}$$

Matrix addition is commutative, i.e. **A** + **B** = **B** + **A**.

$$(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T \tag{338}$$

- **Hermitian Transpose:** When the Hermitian transpose of a complex matrix is found, the *n*-th row of the matrix is written as the *n*-th column and each (complex) element of the matrix is conjugated. The Hermitian transpose of a matrix $\boldsymbol{A}$ is denoted as $\boldsymbol{A}^H$. Note that the matrix product of $\boldsymbol{AA}^H$ will always produce a real and symmetric matrix.

$$\boldsymbol{A} = \begin{bmatrix} (1+2j) & (-2+j) & (-1+4j) \\ (3+j) & (3+7j) & (1+5j) \end{bmatrix} \quad \Leftrightarrow \quad \boldsymbol{A}^H = \begin{bmatrix} (1-2j) & (3-j) \\ (-2-j) & (3-7j) \\ (-1-4j) & (1-5j) \end{bmatrix}$$
$$\Rightarrow \boldsymbol{AA}^H = \begin{bmatrix} 27 & 25 \\ 25 & 84 \end{bmatrix} \tag{339}$$

Note that if a matrix, $\boldsymbol{B}$, has only real number elements, then $\boldsymbol{B}^H = \boldsymbol{B}^T$. See also *Matrix Properties - Hermitian, Complex Matrix, Matrix.*

- **Inverse:** If for two square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$:

$$\boldsymbol{AB} = \boldsymbol{I} \tag{340}$$

then $\boldsymbol{B}$ can be referred to as the inverse of $\boldsymbol{A}$, or $\boldsymbol{B} = \boldsymbol{A}^{-1}$. If $\boldsymbol{A}^{-1}$ exists, then $\boldsymbol{A}$ is non-singular. Note that

$$\boldsymbol{AA}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I} \tag{341}$$

and

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1} \tag{342}$$

For example

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 3 \\ 0 & 1 & 2 \end{bmatrix} \Rightarrow \quad \boldsymbol{A}^{-1} = \begin{bmatrix} -1 & 1 & -1 \\ -4 & 2 & -1 \\ 2 & -1 & 1 \end{bmatrix} \tag{343}$$

$$\boldsymbol{AA}^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 3 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} -1 & 1 & -1 \\ -4 & 2 & -1 \\ 2 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{344}$$

Inversion of matrices is useful for analytical procedures in DSP, however its use in real time computation is rare because of the very large computation requirements and the potential numerical instability of the algorithm. In general the explicit inversion of matrices is circumvented by the use of linear algebraic

methods such as LU decomposition (with pivoting), QR decomposition and Cholesky decomposition (for symmetric matrices) which have improved numerical properties [15].

- **Kronecker Product:** This is a useful mathematical operator for generating vectors and matrices. It is particularly useful in interpretive programming languages such as Matlab$_{TM}$ for implementing simple DSP operations such as upsampling. In general, the Kronecker Product multiplies every element of one matrix by a second matrix and arranges these matrices into the same shape as the first matrix.

- **Multiplication:** The multiplication of two matrices ***AB*** is only possible when the number of columns in ***A*** is the same as the number of rows in ***B***. Each row of matrix ***A*** is multiplied by each column of ***B*** in a sum of products (or vector inner product) form. If ***A*** is an $m \times n$ matrix and ***B*** is an $n \times p$ matrix the result will be ***C***, an $m \times p$ matrix. (Note that because of the dimensions the product of ***BA*** cannot be formed unless $m = p$. Matrix matrix multiplication is not a commutative operation, i.e. in general $\boldsymbol{AB} \neq \boldsymbol{BA}$)

  For example, if we form the matrix product ***C = AB***, where ***A*** is a 3×4, and ***B*** is a 4×2 matrix:

$$\boldsymbol{A} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} m & n \\ o & p \\ q & r \\ s & t \end{bmatrix} \tag{345}$$

  then

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix} \begin{bmatrix} m & n \\ o & p \\ q & r \\ s & t \end{bmatrix} = \begin{bmatrix} (am+bo+cq+ds) & (an+bp+cr+dt) \\ (me+fo+gq+hs) & (ne+fp+gr+ht) \\ (im+jo+kq+ls) & (in+jp+kr+lt) \end{bmatrix}$$

$$\begin{matrix} \boldsymbol{A} & \boldsymbol{B} & & \boldsymbol{C} \\ 3\times4 & 4\times2 & & 3\times2 \end{matrix}$$

  In general for an $m \times n$ matrix, A, and an $n \times p$ matrix, B, the $m \times p$ elements of the product matrix ***C*** will have elements:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \tag{346}$$

- **Matrix-Vector Multiplication:** Multiplication of a vector by a matrix is a special case of matrix multiplication, where one of the matrices to be multiplied is a vector, or $n \times 1$ matrix. Multiplication of an $n \times 1$ vector by an $m \times n$ vector yields an $m \times 1$.

$$\boldsymbol{y} = \boldsymbol{Rx} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} (a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4) \\ (a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4) \\ (a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4) \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \tag{347}$$

$$3 \times 4 \qquad 4 \times 1 \qquad\qquad 3 \times 1$$

- **Premultiplication:** See *Postmultiplication.*

- **Postmultiplication:** Noting that in general for two matrices, ***A*** and ***B,*** (of dimension $n \times m$ and $m \times n$ respectively):

$$\boldsymbol{AB} \neq \boldsymbol{BA} \tag{348}$$

and therefore when multiplying two matrices it is important to specify the order. If it is required to multiply two matrices then the order can be verbosely described using the term *postmultiplication* or *premultiplication*. To state that matrix $\boldsymbol{C}$ is formed by $\boldsymbol{A}$ being *postmultiplied* by $\boldsymbol{B}$ means:

$$\boldsymbol{C} = \boldsymbol{AB} \tag{349}$$

which is equivalent to stating that $\boldsymbol{B}$ is *premultiplied* by $\boldsymbol{A}$.

- **Scaling:** A matrix, $\boldsymbol{A}$, is scaled by multiplying every element by a scale factor, c.

$$c\boldsymbol{A} = c\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} ca_{11} & ca_{12} & ca_{13} \\ ca_{21} & ca_{22} & ca_{23} \\ ca_{31} & ca_{32} & ca_{33} \end{bmatrix} \tag{350}$$

- **Transpose:** The transpose of a matrix is obtained by writing the n-th column (top to bottom) of the matrix as the n-th row (left to right). The transpose of a matrix, $\boldsymbol{A}$, is denoted as $\boldsymbol{A}^{\mathrm{T}}$. For example, if:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{A}^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \end{bmatrix} \tag{351}$$

Therefore if $\boldsymbol{B} = \boldsymbol{A}^T$, then for every element of $\boldsymbol{A}$ and $\boldsymbol{B}$, $a_{ij} = b_{ji}$. Note also the identity:

$$(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T \tag{352}$$

and

$$(\boldsymbol{A}^T)^T = \boldsymbol{A} \tag{353}$$

The product of $\boldsymbol{A}^T\boldsymbol{A}$ is frequently found in DSP particularly in least squares derived algorithms. See also *Hermitian Transpose.*

- **Subtraction:** See *Matrix-Vector Addition.*
- **Vector-Matrix Multiplication:** See *Matrix-Vector Multiplication.*

Matrix Operation Complexity: The number of arithmetic operations to perform the fundamental matrix operations of addition (subtraction), multiplication and inversion can be given in terms of the number of multiplies, adds, divisions and square roots that are required.

Matrix Operation	Matrix Dimension	Additions	Multiplies	Divides/Sqrts
Addition $\boldsymbol{A} + \boldsymbol{B}$	$(m \times n) + (m \times n)$	mn	0	0
Multiplication $\boldsymbol{AB}$	$(m \times n).(n \times p)$	mnp	mnp	0
Inversion $\boldsymbol{A}^{-1}$	$(n \times n)$	$O(n^3)$	$O(n^3)$	$O(n^2)$

In general if a matrix is sparse (e.g. upper triangular, diagonal etc.) then the number of arithmetic operations will be reduced since operations with one or more zero arguments need not be

performed. For example multiplication of two diagonal matrices both of dimension $n \times n$ requires only n multiplies and adds. Also inversion of a diagonal matrix only requires n divisions.

It is worth noting that the matrix inverse is rarely calculated explicitly and systems of linear equations of the form $\boldsymbol{Ax} = \boldsymbol{b}$ are usually solved via Gaussian Elimination, or QR decomposition type algorithms [15].

Matrix, Partitioning: It is often convenient to group the elements of a matrix into smaller submatrices either for notational convenience or to highlight a logical division between two quantities represented in the same matrix. For example the 6×4 matrix $\boldsymbol{A}$, can be partitioned into four 3×2 submatrices:

$$\boldsymbol{A} = \left[\begin{array}{cc:cc} a_{11} & a_{12} & a_{13} & a_4 \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ \hdashline a_{41} & a_{42} & a_{43} & a_{44} \\ a_{51} & a_{52} & a_{53} & a_{54} \\ a_{61} & a_{62} & a_{63} & a_{64} \end{array}\right] = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \tag{354}$$

A partitioned matrix is often referred to as a block matrix, i.e. a matrix in which the elements are submatrices, rather than scalars. The use of block matrices is often exploited in the development of DSP algorithms for notational convenience.

The specification of an algorithm using partitioned matrices (block matrices) is often referred to as a block algorithm. Block algorithms (such as block matrix multiplication and addition etc.) should be expressed such that the block dimensions and the submatrix dimensions are consistent with the normal procedures of the matrix operation. QR decomposition and the matrix vector form of an IIR filter can be conveniently represented as block matrix algorithms.

For example consider the multiplication of the 6×4 matrix partitioned into 3×2 blocks (or submatrices) by a 4×4 matrix partitioned into 2×2 blocks or submatrices. The product $\boldsymbol{C} = \boldsymbol{AB}$ can be expressed in terms of the submatrices. Note that the dimensions of the submatrices $\boldsymbol{A}_{im}$ and $\boldsymbol{B}_{mj}$ must be such that they can be matrix multiplied. In this example the result gives submatrices $\boldsymbol{C}_{ij}$ of dimension 3×2.

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} \\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22} \end{bmatrix} \tag{355}$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} \\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{A}_{11}\boldsymbol{B}_{11} + \boldsymbol{A}_{12}\boldsymbol{B}_{21}) & (\boldsymbol{A}_{11}\boldsymbol{B}_{12} + \boldsymbol{A}_{12}\boldsymbol{B}_{22}) \\ (\boldsymbol{A}_{21}\boldsymbol{B}_{11} + \boldsymbol{A}_{22}\boldsymbol{B}_{21}) & (\boldsymbol{A}_{21}\boldsymbol{B}_{12} + \boldsymbol{A}_{22}\boldsymbol{B}_{22}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{bmatrix} \tag{356}$$

Matrix Properties: In this entry *properties* of a matrix include useful identities and general forms of information that can be extracted from or stated about a matrix. See also *Matrix Decompositions, Matrix Operations.*

- **Condition Number:** The *condition number* provides a measure of the ill-condition or poor numerical behavior of a matrix. Consider the following set of equations where $\boldsymbol{A}$ is a known $n \times n$ non singular matrix, and $\boldsymbol{b}$ is a known $n \times 1$ vector:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{357}$$

The solution to this system of equations is well known to be:

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{358}$$

Using a processor with "infinite" arithmetic precision an exact answer will be obtained. If however the equation is to be solved using finite precision arithmetic, then this can be modeled as a small error added to the elements of $\boldsymbol{A}$ and $\boldsymbol{d}$ where this error is such that:

$$\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|} \approx \varepsilon \quad \text{and} \quad \frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \approx \varepsilon \quad \text{and} \quad \varepsilon \ll 1 \tag{359}$$

Therefore the problem is now one of solving:

$$\boldsymbol{x} + \delta\boldsymbol{x} = (\boldsymbol{A} + \delta\boldsymbol{A})^{-1}(\boldsymbol{b} + \delta\boldsymbol{b}) \tag{360}$$

where $\delta\boldsymbol{A}$ and $\delta\boldsymbol{b}$ represent the error (or perturbation) matrix and vector of $\boldsymbol{A}$ and $\boldsymbol{b}$ respectively. It can be shown that the relative error of the norm (perturbation) of the vector $\boldsymbol{x}$ is given by:

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \le \varepsilon\kappa(\boldsymbol{A}) \tag{361}$$

where for a square matrix $\boldsymbol{A}$ the *condition number,* $\kappa(\boldsymbol{A})$, is defined as:

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\| \tag{362}$$

The norm of a matrix, $\|\boldsymbol{A}\|$, gives information in some sense of the magnitude of the matrix. One measure of matrix norm is its largest singular value. If the matrix $\boldsymbol{A}$ is decomposed using the singular value decomposition (SVD):

$$\boldsymbol{A} = \boldsymbol{U}\Sigma\boldsymbol{V}^T \tag{363}$$

where $\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_n)$ is a diagonal matrix denoting the singular values of $\boldsymbol{A}$, and $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices. The condition number of matrix $\boldsymbol{A}$, denoted as $\kappa(\boldsymbol{A})$ is defined as the ratio of the largest singular value to the smallest singular value (in accordance with Eq. 362):

$$\kappa(\boldsymbol{A}) = \frac{\max(\sigma_i)}{\min(\sigma_i)} \quad \text{for} \quad 0 < i < n \tag{364}$$

Therefore if a matrix has a very large *condition number* a simple interpretation is that when solving equations of the form in Eq. 358 then even very small errors in the matrix $\boldsymbol{A}$, as modelled in Eq. 360, may lead to very large errors in the solution vector $\boldsymbol{x}$; hence "numerical" care must be taken.

To state the relevance of $\kappa(\boldsymbol{A})$ in another way, if the condition number is very large then this implies that when calculating the inverse matrix:

$$\boldsymbol{A}^{-1} = \boldsymbol{V}\Sigma^{-1}\boldsymbol{U}^T \tag{365}$$

the dynamic range of numbers in the inverse will be very large. This easily seen by noting that $\Sigma = \text{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \sigma_3^{-1}, \ldots, \sigma_n^{-1})$. For example if:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad \text{then} \quad \boldsymbol{A}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \quad \text{and} \quad \kappa(\boldsymbol{A}) = 2 \tag{366}$$

the matrix $\boldsymbol{A}$ is well-conditioned and a numerical dynamic range of around 0.1 to 10 ($40\text{dB} = 20\log(10/0.1)$) is "suitable" for the arithmetic. However for a matrix $\boldsymbol{B}$:

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 \\ 0 & 0.0001 \end{bmatrix} \quad \text{then} \quad \boldsymbol{B}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 10000 \end{bmatrix} \quad \text{and} \quad \kappa(\boldsymbol{B}) = 10000 \tag{367}$$

the condition number highlights the ill-conditioning of the matrix, and this time a numerical dynamic range of around 0.00001 to 10000 (160dB) is required for reliable arithmetic. Therefore matrix $\boldsymbol{A}$ could be reliably inverted by a 16 bit DSP processor (96dB dynamic range), whereas matrix $\boldsymbol{B}$ would require a 32 bit floating point DSP processor (764dB dynamic range).

Note that the larger the condition number the "closer" the matrix is to singularity. A singular matrix will have a condition number of ∞.

For analysis of many DSP algorithms note that the condition number is often given as the ratio of the largest eigenvalue to the smallest eigenvalue:

$$\kappa(\boldsymbol{A}) = \frac{\text{Largest Eigenvalue}}{\text{Smallest Eigenvalue}} = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{368}$$

This is because in most DSP problems solved using linear algebra techniques the matrix $\boldsymbol{A}$ is square and very often symmetric positive definite, and the eigenvalue decomposition is in fact a special case of the more general singular value decomposition, and the eigenvalues are the same as the singular values. See also *Adaptive Signal Processing, Matrix Decompositions - Eigenvalue/Singular Value, Matrix Properties - Norm/Eigenvalue Ratio, Vector Properties - Norm, Recursive Least Squares.*

- **Conjugate Transpose:** See *Matrix Properties - Hermitian Transpose.*
- **Determinant:** Noting that the for a 1×1 matrix, $\alpha = [a]$, the *determinant* is given by $\det(\alpha) = a$, the *determinant* of a square matrix, $\boldsymbol{A}$ of dimension $m \times m$ can be defined recursively in terms of the *determinant* of a related $(m-1) \times (m-1)$ matrix, $\boldsymbol{A}_{1i}$ obtained by deleting the first row and the *i*-th column of $\boldsymbol{A}$.

$$\det(\boldsymbol{A}) = \sum_{i=1}^{m} (-1)^{i+1} a_{1i} \det(\boldsymbol{A}_{1i}) \tag{369}$$

where a_{1i} is the first element in the *i*-th column of the matrix. If $\det(\boldsymbol{A}) = 0$ then the matrix is singular. Also for two square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ it can be shown that $\det(\boldsymbol{AB}) = \det(\boldsymbol{A})\det(\boldsymbol{B})$, and $\det(\boldsymbol{A}^T) = \det(\boldsymbol{A})$. In general the determinant of a matrix defines the number of independent rows/ columns of the matrix.

- **Eigenvalue:** For a square $n \times n$ matrix, $\boldsymbol{A}$, if there exists a non-zero $n \times 1$ vector $\boldsymbol{x}$, and a non-zero scalar λ such that:

$$\boldsymbol{Ax} = \lambda \boldsymbol{x} \tag{370}$$

then λ is an eigenvalue and $\boldsymbol{x}$ is an eigenvector of matrix $\boldsymbol{A}$. See also *Matrix Decompositions - Eigenanalysis.*

- **Eigenvalue Ratio:** The ratio of the largest eigenvalue to the smallest eigenvalue, denoted $\kappa(\boldsymbol{A})$, for a square symmetric positive definite matrix, $\boldsymbol{A}$:

$$\kappa(\boldsymbol{A}) = \frac{\text{Largest Eigenvalue}}{\text{Smallest Eigenvalue}} = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{371}$$

is more precisely known as the *condition number* of a matrix. The eigenvalue ratio (also known as eigenvalue spread) gives information about the general numerical behavior (good or otherwise!) of a data matrix $\boldsymbol{A}$ to when a problem usually of the form, $\boldsymbol{Ax} = \boldsymbol{b}$ is solved for the unknown vector $\boldsymbol{x}$, i.e. $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$. See also *Matrix Properties - Condition Number, Matrix Decompositions - Eigenvalue/Singular Value, Adaptive Signal Processing Algorithms.*

- **Eigenvalue Spread:** See entry for *Matrix Properties - Condition Number/Eigenvalue Ratio.*

- **Frobenius Norm:** See *Matrix Properties - Norm.* See also *Vector Properties - Norm.*
- **Hermitian (Symmetric):** A complex matrix is often described as *Hermitian* if $\boldsymbol{A} = \boldsymbol{A}^H$. Synonymous names are *Hermitian symmetric*, or complex-symmetric. Note that if the matrix $\boldsymbol{A}$ is real, then $\boldsymbol{A} = \boldsymbol{A}^T$ and $\boldsymbol{A}$ would be described as symmetric. See *Matrix Decompositions - Hermitian Transpose.*
- **Hermitian Transpose:** For two complex matrices $\boldsymbol{A}$ $(m \times n)$ and $\boldsymbol{B}$ $(n \times m)$ then the *Hermitian transpose* of the product can be written as:

$$(\boldsymbol{AB})^H = \boldsymbol{B}^H\boldsymbol{A}^H \tag{372}$$

Note that:

$$(\boldsymbol{A}^H)^H = \boldsymbol{A} \tag{373}$$

A "dagger" is often used as the Hermitian transpose symbol, i.e. $\boldsymbol{A}^H = \boldsymbol{A}^\dagger$

The matrix product $\boldsymbol{R}$ of an $m \times n$ matrix, $\boldsymbol{A}$ and its Hermitian transpose, $\boldsymbol{A}^H$ will always produce a conjugate symmetric $m \times m$ matrix, i.e. $\boldsymbol{R} = \boldsymbol{R}^H$:

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} (1+2j) & (-2+j) & (-1+4j) \\ (3+j) & (3+7j) & (1+5j) \end{bmatrix} \quad \Leftrightarrow \quad \boldsymbol{A}^H = \begin{bmatrix} (1-2j) & (3-j) \\ (-2-j) & (3-7j) \\ (-1-4j) & (1-5j) \end{bmatrix} \\ \Rightarrow \boldsymbol{R} &= \boldsymbol{A}\boldsymbol{A}^H = \begin{bmatrix} 27 & 25+31j \\ 25-31j & 84 \end{bmatrix} = \boldsymbol{R}^H \end{aligned} \tag{374}$$

(also, if $\boldsymbol{A}$ is full rank, the $\boldsymbol{R}$ will be positive definite, otherwise $\boldsymbol{R}$ will be positive semi-definite).

Note that if a matrix, $\boldsymbol{B}$, has only real number elements, then the *Hermitian transpose* is equivalent to the normal matrix transpose, i.e. $\boldsymbol{B}^H = \boldsymbol{B}^T$. See also *Complex Matrix, Complex Numbers, Matrix Properties - Hermitian Transpose.*

- **Ill-Conditioned:** An $m \times n$ matrix, $\boldsymbol{A}$ is said to be *ill-conditioned* when the condition number, calculated as the ratio of the maximum singular value to minimum singular value (or maximum eigenvalue to minimum eigenvalue for $n \times n$ matrices) is very high. A matrix that is not *ill-conditioned* is well-conditioned. For more detail see entry *Matrix Properties - Condition Number.* See also *Matrix Decompositions - Eigenvalue/Singular Value*
- $.\infty$ **-norm:** See *Matrix Properties - Norm.*
- **Inversion:** For two square invertible matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ then:

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1} \tag{375}$$

See also *Matrix Operations - Inversion.*

- **Inversion Lemma:** If $\boldsymbol{A}$ and $\boldsymbol{C}$ are nonsingular square matrices and $\boldsymbol{B}$ and $\boldsymbol{D}$ are of compatible dimension such that:

$$\boldsymbol{P} = \boldsymbol{A} + \boldsymbol{BCD} \tag{376}$$

and $\boldsymbol{P}$ is non singular, then the *matrix inversion lemma* allows $\boldsymbol{P}^{-1}$ to be expressed as:

$$\boldsymbol{P}^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{C}^{-1} + \boldsymbol{D}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{D}\boldsymbol{A}^{-1} \tag{377}$$

This identity can be confirmed by multiplying the right sides of Eq. 376 and Eq. 377 together:

$$
\begin{aligned}
&(\boldsymbol{A}+\boldsymbol{BCD})(\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B})^{-1}\boldsymbol{DA}^{-1}) \\
&= \boldsymbol{I}+\boldsymbol{BCDA}^{-1}-\boldsymbol{B}(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B})^{-1}\boldsymbol{DA}^{-1}+\boldsymbol{BCDA}^{-1}\boldsymbol{B}(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B})^{-1}\boldsymbol{DA}^{-1} \\
&= \boldsymbol{I}+\boldsymbol{BCDA}^{-1}-\boldsymbol{B}[(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B})^{-1}+\boldsymbol{CDA}^{-1}\boldsymbol{B}(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B})^{-1}]\boldsymbol{DA}^{-1} \\
&= \boldsymbol{I}+\boldsymbol{BCDA}^{-1}-\boldsymbol{B}(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B})^{-1}(\boldsymbol{I}+\boldsymbol{CDA}^{-1}\boldsymbol{B})\boldsymbol{DA}^{-1} \\
&= \boldsymbol{I}+\boldsymbol{BCDA}^{-1}-\boldsymbol{B}(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B})^{-1}(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B})\boldsymbol{CDA}^{-1} \\
&= \boldsymbol{I}+\boldsymbol{BCDA}^{-1}-\boldsymbol{BCDA}^{-1} \\
&= \boldsymbol{I} \qquad \text{QED}
\end{aligned}
\tag{378}
$$

For some digital signal processing algorithms (such as the recursive least squares (RLS) algorithm) it is often that case that $\boldsymbol{C}$ is a 1×1 identity matrix, $\boldsymbol{B}$ is a vector and $\boldsymbol{D}$ the same vector transposed. Also for notational reasons $\boldsymbol{A}$ is written as an inverse matrix. Therefore applying the matrix inversion lemma to:

$$\boldsymbol{P} = \boldsymbol{R}^{-1}+\boldsymbol{vv}^{T} \tag{379}$$

gives

$$\boldsymbol{P}^{-1} = \boldsymbol{R}-\boldsymbol{Rv}(1+\boldsymbol{v}^{T}\boldsymbol{Rv})\boldsymbol{v}^{T}\boldsymbol{R} \tag{380}$$

- **Non-negative Definite:** See entry for *Matrix Properties - Positive Definite.*
- **Nonsingular:** See *Matrix Properties - Singular.*
- **Norm:** A matrix norm gives a measure of the overall magnitude of the matrix space. The most common norms are the *Frobenius norm* and the set of *p-norms*.

  The *Frobenius norm* of an $m\times n$ matrix $\boldsymbol{A}$, is usually denoted, $\|\boldsymbol{A}\|_F$ and calculated as:

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2} \tag{381}$$

  The *p-norms* are generally defined in terms of vector *p-norms* and calculated as

$$\|\boldsymbol{A}\|_p = \max\frac{\|\boldsymbol{Ax}\|_p}{\|\boldsymbol{x}\|_p} \tag{382}$$

  This can also be expressed in the form:

$$\|\boldsymbol{A}\|_p = \max\|\boldsymbol{Au}\|_p \quad \text{where} \quad \|\boldsymbol{u}\|_p = 1 \tag{383}$$

  On an intuitive level, the matrix 2-norm gives information on the amount by which a matrix will "amplify" the length (vector 2-norm) of any unit vector. Typically $p = 1, 2$ or ∞. Note that the ∞ norm is easily calculated as the largest element magnitude in a matrix. See also *Matrix Properties - Condition Number, Vector Properties - Norms.*
- **Null Space:** The *null space* of $\boldsymbol{A}$ is defined as:

$$\text{null}(\boldsymbol{A}) = \{\boldsymbol{x}\in \Re^{N}, \quad \text{where} \quad \boldsymbol{Ax} = \boldsymbol{0}\} \tag{384}$$

  Intuitively, the null space of $\boldsymbol{A}$ is the set of all vectors orthogonal to the rows of $\boldsymbol{A}$. See also *Matrix Properties - Rank/Range, Vector Properties - Space/Subspace.*
- **1-norm:** See *Matrix Properties - Norm.*
- **Overdetermined System:** The linear set of equations, $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A}$ is a known $m\times n$ matrix with linearly independent columns (i.e. $\text{rank}(\boldsymbol{A}) = n$), $\boldsymbol{b}$ is a known m element vector and $\boldsymbol{x}$ is an unknown n element vector, is said to be overdetermined if $m > n$ thus meaning there are more equations than unknowns. An overdetermined system of equations has no exact solution for $\boldsymbol{x}$. However by minimizing

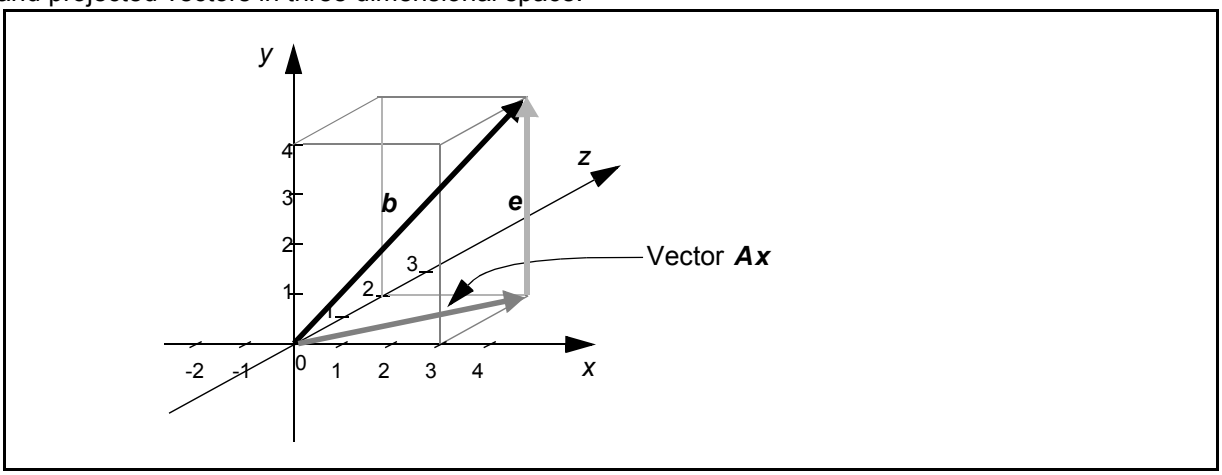
the 2-norm of the error vector, $\boldsymbol{e} = \boldsymbol{Ax} - \boldsymbol{b}$ i.e. minimizing $\varepsilon = (\|\boldsymbol{Ax} - \boldsymbol{b}\|_2)$, the least squares solution is found:

$$\boldsymbol{x}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{385}$$

For example, given the overdetermined system of equations (note there is no exact solution):

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 2 \end{bmatrix} \tag{386}$$

we can make a geometrical interpretation of the least squares solution by representing the various vectors and projected vectors in three dimensional space:



Now considering the subspace defined by the matrix:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{387}$$

the columns only span the x-z plane ($y = 0$) of the above three-dimensional space. Therefore the vector $\boldsymbol{Ax}_{LS}$ that minimizes the norm of the error vector, $\varepsilon = \|\boldsymbol{Ax} - \boldsymbol{b}\|_2$, must lie on the x-z plane. Using the least squares solution:

$$\boldsymbol{x}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} = \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \right]^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \tag{388}$$

From the above geometrical representation it should be clear that the because the vector $\boldsymbol{Ax}$ is constrained to lie in the x-z plane, if the 2-norm (Euclidean length) of the error vector $\boldsymbol{e} = \boldsymbol{Ax} - \boldsymbol{b}$ is to be minimized this will occur when $\boldsymbol{e}$ is perpendicular (orthogonal) to the x-z plane, i.e. the same solution as the least squares. For problems with more than three dimensions a geometric interpretation cannot be offered explicitly, however intuition gained from simpler examples is useful. See also *Least Squares, Square System of Equations, Matrix Properties - Underdetermined System, Vector Properties - 2 norm.*

- **Positive Definite:** An $n \times n$ square matrix, $\boldsymbol{A}$, is *positive definite* if:

$$\boldsymbol{x}^T\boldsymbol{Ax} > 0 \tag{389}$$

for all non-zero n element vectors, $\boldsymbol{x}$.

If

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \geq 0 \tag{390}$$

then $\boldsymbol{A}$ is said to be positive semi-definite or non-negative definite.

Note that if a matrix $\boldsymbol{B}$ has full column rank, then a matrix $\boldsymbol{R}$, calculated as $\boldsymbol{R} = \boldsymbol{B}^T\boldsymbol{B}$ is always positive definite. $\boldsymbol{R}$ will also be symmetric. This can be simply seen by noting that:

$$\boldsymbol{x}^T\boldsymbol{R}\boldsymbol{x} = \boldsymbol{x}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{x} = \|\boldsymbol{B}\boldsymbol{x}\|_2^2 \tag{391}$$

where $\|\boldsymbol{B}\boldsymbol{x}\|_2^2$ is the square of the 2-norm of the vector which is always a positive quantity for non zero vectors $\boldsymbol{x}$. Noting that a symmetric matrix can always be decomposed into its square root or Cholesky form, then all symmetric matrices are positive definite. See also *Correlation Matrix, Matrix Decompositions - Cholesky, Vector Properties - Norm.*

- **Positive Semi-definite:** See entry for *Matrix Properties - Positive Definite.*
- **Pseudo-Inverse:** If an $m \times n$ matrix, where $m > n$ has rank($\boldsymbol{A}$) = n, then the system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ cannot be solved by calculating $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ because $\boldsymbol{A}$ is clearly non-invertible. However the least squares solution can be found such that:

$$\boldsymbol{x}_{LS} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{392}$$

If $\boldsymbol{A}$ is not full rank (i.e., rank($\boldsymbol{A}$)<n) however, then the inverse of $(\boldsymbol{A}^T\boldsymbol{A})$ will fail to exist. In this case, the pseudo-inverse of $\boldsymbol{A}$, $\boldsymbol{A}^+$, is used. The *pseudo-inverse* is defined from the singular value decomposition of $\boldsymbol{A}$ as:

$$\boldsymbol{A}^+ = \boldsymbol{V}\begin{bmatrix}\Sigma^{-1} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{U}^T \tag{393}$$

where $\boldsymbol{A}$ has been decomposed *(see Matrix Decompositions-Singular Value)* into

$$\boldsymbol{A} = \boldsymbol{U}\begin{bmatrix}\Sigma & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{V}^T \tag{394}$$

with Σ being a rank r (r<n) diagonal matrix with a well-defined inverse. If $\boldsymbol{A}$ happens to be full rank then the *pseudo-inverse* can be directly related to $\boldsymbol{A}$ as: $\boldsymbol{A}^+ = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$ if m>n. While we have focussed on the over-determined problem here, we should note that the *pseudoinverse* also provides a minimum norm solution for the underdetermined problem where $\boldsymbol{A}$ is rank deficient.

See also *Least Squares, Matrix Decompositions - Singular Value Decomposition, Overdetermined System, Underdetermined System.*

- **Rank:** The *rank* of a matrix is equal to the number of independent rows or columns of the matrix. For an $m \times n$ matrix, $\boldsymbol{A}$, where $m \geq n$, then $\text{rank}(\boldsymbol{A}) = n$ if and only if the column vectors are linearly independent; note than $\text{rank}(\boldsymbol{A}) = \text{rank}(\boldsymbol{A}^T)$. Similarly, if $m < n$ then $\text{rank}(\boldsymbol{A}) = m$ if and only if the row vectors of A are linearly independent. If $\text{rank}(\boldsymbol{A}) < \min(m, n)$ then the matrix may be described as rank deficient. Note that for an $m \times m$ square matrix, if $\text{rank}(\boldsymbol{A}) < m$ then the matrix is singular.

  While in an analytical, academic framework (i.e., infinite precision), the concept of rank is clearly defined, it becomes somewhat more problematic to define rank when working with matrix based packages such as Matlab$_{TM}$. Because of round-off errors, it is possible to have a test for matrix rank indicate a full rank matrix, when the matrix is actually very poorly conditioned. In some cases software packages warn of rank deficiencies (especially on matrix inversions). However, in DSP applications the significance of low power dimensions is often very application specific. Therefore, it is generally a good idea to pay attention to the condition number of matrices with which you are working. As an example, if you are performing a least squares filter design and the coefficient magnitudes become enormous (say on the order of 10^{15}) when you were expecting much more reasonable numbers (say 10^{-1}, 10^{1}, etc.) this is a good indication of possible rank deficiency (in this case, the rank deficiency is unlikely to be detected by software monitoring).

See also *Matrix Properties - Range/Singular/Condition Number.*

- **Rank Deficient:** See entry for *Matrix Properties - Rank.*
- **Range Space:** For an $m \times n$ matrix $\boldsymbol{A}$, the subspace spanned by the column partitioning of the matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \ldots, \boldsymbol{a}_n]$ is referred to as the *range space* of the matrix. Therefore:

$$\text{range}(\boldsymbol{A}) = \{\boldsymbol{y} \in \Re^m, \text{ where } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}\}, \quad \text{for any } \boldsymbol{x} \in \Re^n \tag{395}$$

See also *Vector Properties - Space/Subspace.*

- **Singular:** For a square matrix, $\boldsymbol{A}$, if there exists no matrix, $\boldsymbol{X}$ such that $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{I}$ (where $\boldsymbol{I}$ is the identity matrix) then the inverse matrix, $\boldsymbol{A}^{-1}$ does NOT exist and the matrix is singular; otherwise the matrix is nonsingular. For example the matrix:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 9 & 0 \end{bmatrix} \tag{396}$$

is singular as there exists no matrix $\boldsymbol{X}$ such that $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{I}$. For an $n \times n$ singular matrix, $\boldsymbol{A}$, the rank will less than n. See also *Matrix Decompositions - Singular Value Decompositions, Matrix Properties - Pseudo-Inverse.*

- **Singular Value:** See *Matrix Decompositions - Singular Value Decomposition.*
- **Sherman-Morrison-Woodbury Formula:** See *Matrix Properties - Inversion Lemma.*
- **Space:** See *Vector Properties - Space.*
- **Square Root Matrix:** If a symmetric matrix, $\boldsymbol{R}$, is decomposed into its Cholesky factors:

$$\boldsymbol{R} = \boldsymbol{L}\boldsymbol{L}^T \tag{397}$$

where $\boldsymbol{L}$ is a lower triangular matrix, $\boldsymbol{L}$ is often also called a *square root matrix* of $\boldsymbol{R}$. There are many other definitions of *matrix square root*. For example, for the symmetric square matrix $\boldsymbol{R}$:

$$\boldsymbol{R}^{\frac{1}{2}} \equiv \boldsymbol{V}\Lambda^{\frac{1}{2}}\boldsymbol{V}^T \tag{398}$$

where the eigen-decomposition of $\boldsymbol{R}$ is used and the square root of the diagonal matrix of eigenvalues is simply defined as the diagonal matrix of the square root of the individual eigenvalues.

See also *Matrix Decompositions - Cholesky/Eigenanalysis.*

- **Square System of Equations:** The linear set of equations:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{399}$$

where $\boldsymbol{A}$ is a known non-singular $n \times n$ matrix (i.e., rank($\boldsymbol{A}$)=n), $\boldsymbol{b}$ is a known n element vector, and $\boldsymbol{x}$ is an unknown n element vector, represents a square system of equations which has an exact solution for $\boldsymbol{x}$ given by:

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{400}$$

For example:

$$\begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ -1 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} -5 \\ 8 \end{bmatrix} \tag{401}$$

For large n it is usually not advisable to calculate $\boldsymbol{A}^{-1}$ directly due to potential numerical instabilities particularly if $\boldsymbol{A}$ is ill-conditioned. Equations of the form in Eq. 399 are best solved using orthogonal techniques such as the QR algorithm, or more general matrix decomposition techniques such LU decomposition (with pivoting), or Cholesky decomposition if $\boldsymbol{A}$ is symmetric. If matrix $\boldsymbol{A}$ has $m > n$ then

the problem is overdetermined and if $m < n$ then the problem is underdetermined. If the rank of $\boldsymbol{A}$ is less than n, then the pseudo-inverse is required. See also *Least Squares, Matrix Decompositions - Cholesky/ LU/QR/SVD, Matrix Properties - Ill-Conditioned/Overdetermined System/Pseudo-Inverse/ Underdetermined System.*

- **Subspace:** See *Vector Properties - Subspace.*
- **Trace:** The trace of a square $n \times n$ matrix, $\boldsymbol{A}$, is defined as the sum of the diagonal elements of that matrix:

$$\text{trace}(\boldsymbol{A}) = \text{trace}\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \sum_{i=1}^{n} a_{ii} \tag{402}$$

It is relatively straightforward (using matrix decompositions) to show that for any $m \times n$ matrix $\boldsymbol{A}$, and any $n \times m$ matrix $\boldsymbol{B}$, then:

$$\text{trace}(\boldsymbol{AB}) = \text{trace}(\boldsymbol{BA}) \tag{403}$$

In DSP a particularly useful property of the trace is that $\text{trace}(\boldsymbol{A}) = \lambda_1 + \lambda_2 + \ldots + \lambda_n$, where λ_i is the i-th eigenvalue of an $n \times n$ matrix $\boldsymbol{A}$. See also *Matrix Decompositions - Eigenanalysis*.

- **Transpose:** For two matrices $\boldsymbol{A}$ ($m \times n$) and $\boldsymbol{B}$ ($n \times m$) then the transpose of the product can be written as:

$$(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T \tag{404}$$

Note that:

$$(\boldsymbol{A}^T)^T = \boldsymbol{A} \tag{405}$$

The product of any $m \times n$ matrix and its transpose gives an $m \times m$ square symmetric matrix:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & -3 \\ 4 & -1 & 5 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{AA}^T = \begin{bmatrix} 1 & 2 & -3 \\ 4 & -1 & 5 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 2 & -1 \\ -3 & 5 \end{bmatrix} = \begin{bmatrix} 14 & -13 \\ -13 & 42 \end{bmatrix} \tag{406}$$

- **2-norm:** See *Matrix Properties - Norm.*
- **Underdetermined System:** The linear set of equations $\boldsymbol{Ax} = \boldsymbol{b}$ is said to be underdetermined, when $\boldsymbol{A}$ is a known $m \times n$ matrix with $m < n$, $\boldsymbol{b}$ is a known m element vector and $\boldsymbol{x}$ is an unknown n element vector. Essentially, there are fewer equations than unknowns and an infinite number of solutions for $\boldsymbol{x}$ exist. If $\boldsymbol{A}$ has linearly independent rows (i.e. $\text{rank}(\boldsymbol{A}) = m$), then there are an infinite number of exact solutions. If rank($\boldsymbol{A}$)<m, however, then the set of equations may be inconsistent, i.e., no exact solution exists. In this latter case, an infinite number of least squares (inexact) solutions exists, with the pseudo-inverse giving the minimum norm solution.

  An underdetermined system of equations has an infinite number of solutions for $\boldsymbol{x}$. Consider the following underdetermined system of equations:

$$\begin{bmatrix} a_{11} & a_{21} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \end{bmatrix}$$
$$\text{i.e.} \quad \begin{bmatrix} a_{11}x_1 + a_{21}x_2 \end{bmatrix} = \begin{bmatrix} b_1 \end{bmatrix} \tag{407}$$

  Choosing *any* value for x_1, a value of x_2 satisfying the underdetermined system of equations can be produced. Hence there is no unique solution and there are an infinite number of solutions. However some

solutions are "better" than others, and the minimum norm solution, where the smallest magnitude 2-norm $\|\boldsymbol{x}\|_2$ is calculated can be found using least squares techniques.

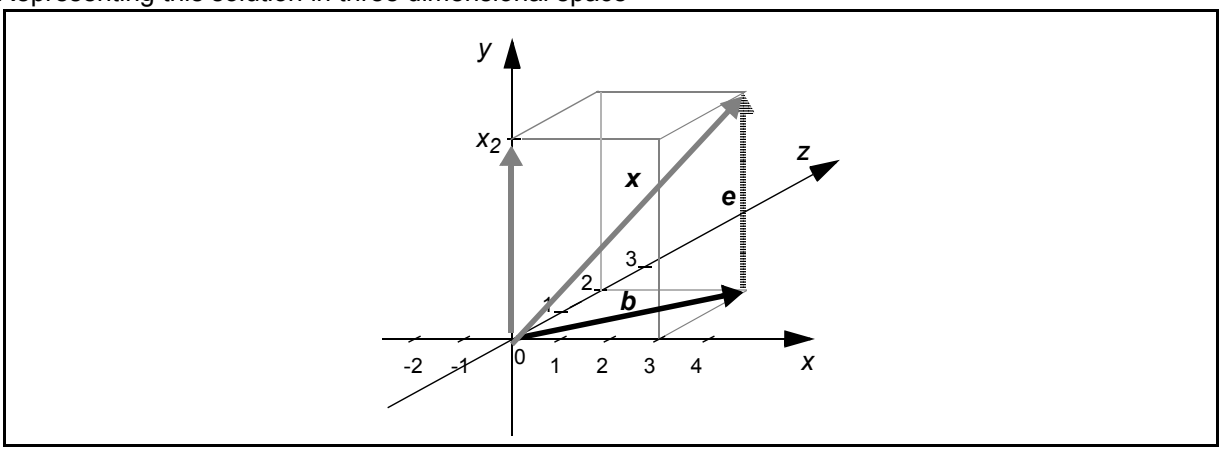
The overdetermined problem can be usefully illustrated geometrically. Consider the following overdetermined system of equations:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \tag{408}$$

The solution set to Eq. 408 is:

$$x_1 = 3, \quad x_2 = \text{Any Real Number}, \quad x_3 = 2 \tag{409}$$

Representing this solution in three dimensional space



From a geometrical interpretation, regardless of the magnitude of x_2, the matrix $\boldsymbol{A}$ will project the vector $\boldsymbol{x}$ onto $\boldsymbol{b}$.

The underdetermined least squares problem can however be uniquely solved using the minimum norm solution. If the 2-norm of the error vector $\boldsymbol{e} = \boldsymbol{Ax} - \boldsymbol{b}$ is minimized, i.e. $\varepsilon = \|\boldsymbol{e}\|_2 = \|\boldsymbol{Ax} - \boldsymbol{b}\|_2$, then from the above geometrical interpretation the *best* solution occurs when $x_2 = 0$. This solution is unique and best in the sense that the $\boldsymbol{x}$ vector has minimum norm. This solution can be calculated by using the least squares solution for underdetermined systems:

$$\boldsymbol{x}_{LS} = \boldsymbol{A}^T(\boldsymbol{AA}^T)^{-1}\boldsymbol{b} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 2 \end{bmatrix} \tag{410}$$

See also *Least Squares, Matrix Decompositions - Singular Value, Overdetermined systems, Square System of Equations.*

- **Well-Conditioned:** An $m \times n$ matrix, $\boldsymbol{A}$ is said to be *well-conditioned* when the condition number, calculated as the ratio of the maximum singular value to minimum singular value (or maximum eigenvalue to minimum eigenvalue for $n \times n$ matrices) is low relative to the precision of the system on which the matrix is being manipulated. A matrix that is not *well-conditioned* is ill-conditioned. For more details see entry *Matrix Properties - Condition Number.* See also *Matrix Decompositions - Eigenvalue/Singular Value.*
- **Woodbury's Identity**: See *Matrix Properties - Inversion Lemma*.

Matrix Scaling: See *Matrix Operations - Scaling.*

Matrix, Structured: A matrix that has regularly grouped elements and a specific structure of zero elements is called a *structured matrix*. When structured matrices are to be used in calculations, the zeroes in the structure can often be exploited to reduce the total number of computations, and the matrix storage requirements. A number of key structured matrices often found in linear algebra based DSP algorithms and analysis can be identified. See also *Matrix Decompositions, Matrix Operations, Matrix Properties*.

- **Band:** In a band matrix the upper right and lower left corners of the matrix are zero elements, and a band of diagonal elements are non-zero. For example a 5×6 matrix with band width of 3 may have the form:

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & 0 & 0 & 0 & 0 \\ b_{21} & b_{22} & b_{23} & b & 0 & 0 \\ 0 & b_{32} & b_{33} & b_{34} & 0 & 0 \\ 0 & 0 & b_{43} & b_{44} & b_{45} & 0 \\ 0 & 0 & 0 & b_{54} & b_{55} & b_{56} \end{bmatrix} \tag{411}$$

- **Bidiagonal:** A matrix where only the main diagonal, and the first diagonal (above or below the main) are non-zero. See also *Bidiagonalization.*

$$\boldsymbol{E} = \begin{bmatrix} d_1 & g_1 & 0 & 0 \\ 0 & d_2 & g_2 & 0 \\ 0 & 0 & d_3 & g_3 \\ 0 & 0 & 0 & d_4 \end{bmatrix} \tag{412}$$

- **Circulant:** An $n \times n$ *circulant matrix* has only *N* distinct elements, where each row is formed by shifting the previous row by one element to the right in a circular buffer fashion. One interesting property of circulant matrices is that the eigenvalues can be determined by taking a DFT of the first row. The eigenvectors are given by the standard basis vectors of the DFT. See also *Matrix-Structured-Toeplitz.*

$$\boldsymbol{C} = \begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_3 & r_0 & r_1 & r_2 \\ r_2 & r_3 & r_0 & r_1 \\ r_1 & r_2 & r_3 & r_0 \end{bmatrix} \tag{413}$$

- **Diagonal:** A diagonal matrix has all elements, except those on the main diagonal, equal to zero. Multiplying an appropriately dimensioned matrix by a diagonal matrix is equivalent to multiplying the *i*-th row of the matrix, by the *i*-th diagonal element. Diagonal matrices are usually square matrices, although this is not necessarily the case.

$$\boldsymbol{D} = \begin{bmatrix} d_{11} & 0 & 0 & 0 \\ 0 & d_{22} & 0 & 0 \\ 0 & 0 & d_{33} & 0 \\ 0 & 0 & 0 & d_{44} \end{bmatrix} \tag{414}$$

For shorthand, a diagonal matrix is often denoted as:

$\boldsymbol{D}$ = diag(d_1 d_2 d_3 d_4) where $d_i = d_{ii}$.

- **Identity:** The identity matrix has all elements zero, except for the main diagonal elements which are equal to one. The identity matrix is almost universally denoted as $\boldsymbol{I}$. For any matrix $\boldsymbol{A}$, multiplied by the appropriately dimensioned identity matrix, the result is $\boldsymbol{A}$. Any matrix multiplied by its inverse, gives the identity matrix. See also *Diagonal Matrix, Matrix Inverse.*

$$\boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{415}$$

- **Lower Triangular:** A matrix where all elements below the main diagonal are equal to zero. Lower triangular matrices are useful in solving linear algebraic equations with algorithms such as LU (lower, upper) decomposition. Useful properties are that the product of a lower triangular matrix, and a lower triangular matrix is a lower triangular matrix, and the inverse of a lower triangular matrix is a lower triangular matrix. See also *Forward-substitution, Upper Triangular Matrix.*

$$\boldsymbol{L} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \tag{416}$$

- **Orthogonal:** A matrix is called *orthogonal* (or orthonormal) if its transpose, $\boldsymbol{Q}^{\mathrm{T}}$, forms the inverse matrix $\boldsymbol{Q}^{-1}$, i.e. $\boldsymbol{Q}^T = \boldsymbol{Q}^{-1}$ and,

$$\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I} = \boldsymbol{Q}\boldsymbol{Q}^T \tag{417}$$

  It can also be said that the columns of the matrix $\boldsymbol{Q}$ form an orthonormal basis for the space $\Re^m$. While the terms orthogonal and orthonormal are used interchangeably as applied to matrices, they have distinct meanings when applied to sets of functions or vectors -- with orthonormal indicating unit norm for every element in an orthognonal set. See also *Matrix Decompositions Eigenvalue/QR, Matrix Properties - Unitary Matrix.*

- **Orthonormal:** See *Orthogonal.*

- **Permutation:** A matrix that is essentially the identity matrix with the row orders changed. Multiplying another matrix, $\boldsymbol{A}$, by a permutation matrix, $\boldsymbol{P}$, will swap the row orders of $\boldsymbol{A}$. In general multiplication of a matrix by a permutation matrix does not change any of the fundamental quantities such as eigenvalues, condition number. The permutation matrix is an orthogonal matrix.

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{418}$$

- **Rectangular:** A matrix that does not have the same number of rows and columns.

- **Sparse:** Any matrix with a large proportion of zero elements is often termed a *sparse matrix*. Matrices such as lower triangular, diagonal etc can be described as *structured sparse* matrices. When performing matrix algebra on *sparse matrices*, the number of MACs required is usually greatly reduced over an equivalent operation using the a full populated matrix, given that many null operations are performed, e.g. multiplies and additions that have one or two zero values.

- **Square:** A matrix with the same number of rows as columns. Covariance and correlation matrices are necessarily square.

- **Symmetric:** A matrix is symmetric if $\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}$. The line of symmetry is therefore through the main diagonal. Many matrices used in DSP algorithms are symmetric, such as the correlation matrix.

$$\boldsymbol{S} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{12} & s_{22} & s_{23} & s_{24} \\ s_{13} & s_{23} & s_{33} & s_{34} \\ s_4 & s_{24} & s_{34} & s_{44} \end{bmatrix} \tag{419}$$

- **Toeplitz:** This matrix has constant elements in all diagonals. The correlation matrix of stationary stochastic N element data vector forms an $N \times N$ Toeplitz matrix. See also *Matrix-Circulant, and Correlation Matrix, Covariance Matrix.*

$$\boldsymbol{T} = \begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_{-1} & r_0 & r_1 & r_2 \\ r_{-2} & r_{-1} & r_0 & r_1 \\ r_{-3} & r_{-2} & r_{-1} & r_0 \end{bmatrix} \tag{420}$$

- **Tridiagonal:** A matrix where only the main, first upper and first lower diagonals are non-zero elements.

$$\boldsymbol{T} = \begin{bmatrix} t_{11} & s_{12} & 0 & 0 \\ v_{21} & t_{22} & s_{23} & 0 \\ 0 & v_{32} & t_{33} & s_{34} \\ 0 & 0 & v_{43} & t_{44} \end{bmatrix} \tag{421}$$

- **Unitary:** A complex data matrix is *unitary* if the transpose of a complex data orthogonal matrix, $\boldsymbol{U}^{\mathrm{T}}$, forms the inverse matrix $\boldsymbol{U}^{-1}$, i.e. $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$ and therefore,

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} = \boldsymbol{U}\boldsymbol{U}^T \tag{422}$$

  The unitary property is the complex matrix equivalent property of orthogonality. See also *Eigenvalue Decomposition, QR algorithm, Unitary Matrix.*

- **Upper Triangular:** A matrix where all elements above the main diagonal are equal to zero. Upper triangular matrices are useful in solving linear algebraic equations with algorithms such as LU (lower, upper) decomposition. Use properties are that the product of an upper triangular matrix, and an upper triangular matrix is an upper triangular matrix, and the inverse of an upper triangular matrix is an upper triangular matrix. See also *Back-substitution, Lower Triangular Matrix.*

$$\boldsymbol{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} \tag{423}$$

Matrix-Vector Multiplication: See *Matrix Operations - Matrix-Vector Multiplication.*

Maximum Length Sequences: If a binary sequence is produced using a pseudo random binary sequence generator, the sequence is said to be a *maximum length sequence* if for an N bit register, the binary sequence is of length $2^N - 1$ before it repeats itself. In a maximum length sequence the

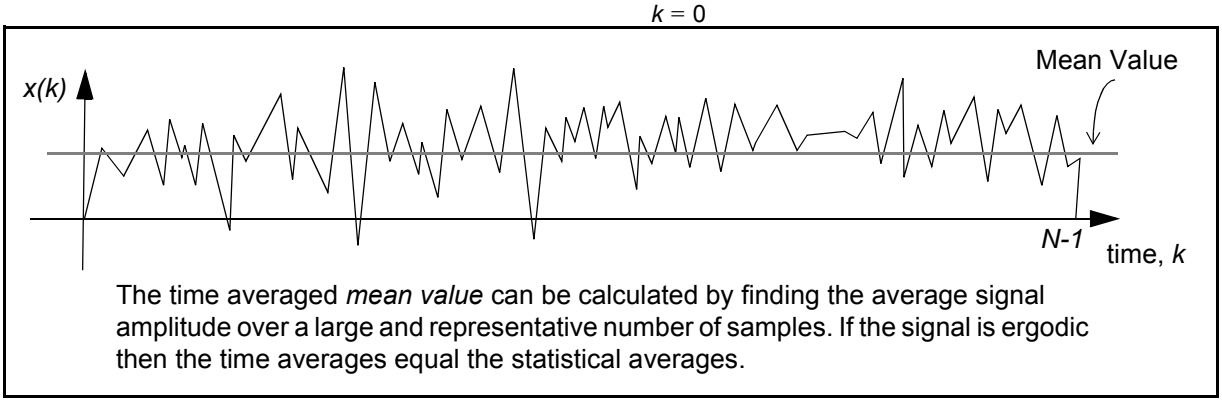
number of 1's is one more than the number of 0's. Also known as m-sequences. See also *Pseudo-Random Binary Sequence*.

Mean Value: The statistical *mean value* of a signal, $x(k)$, is the average amplitude of the signal. Statistical mean is calculated using the statistical expectation operator, $E\{.\}$:

$$E\{x(k)\} = \text{Statistical Mean Value of } x(k) = \sum_{k} x(k)p\{x(k)\} \tag{424}$$

where $p\{x(k)\}$ is the probability density function of $x(k)$. In real time DSP the probability density function of a signal is rarely known. Therefore to find the *mean value* of a signal the more intuitively obvious calculation of a time average computed over a large and representative number of samples, *N*, is used:

$$\text{Time Average} = \frac{1}{N}\sum_{k=0}^{N-1} x(k) \tag{425}$$



The time averaged *mean value* can be calculated by finding the average signal amplitude over a large and representative number of samples. If the signal is ergodic then the time averages equal the statistical averages.

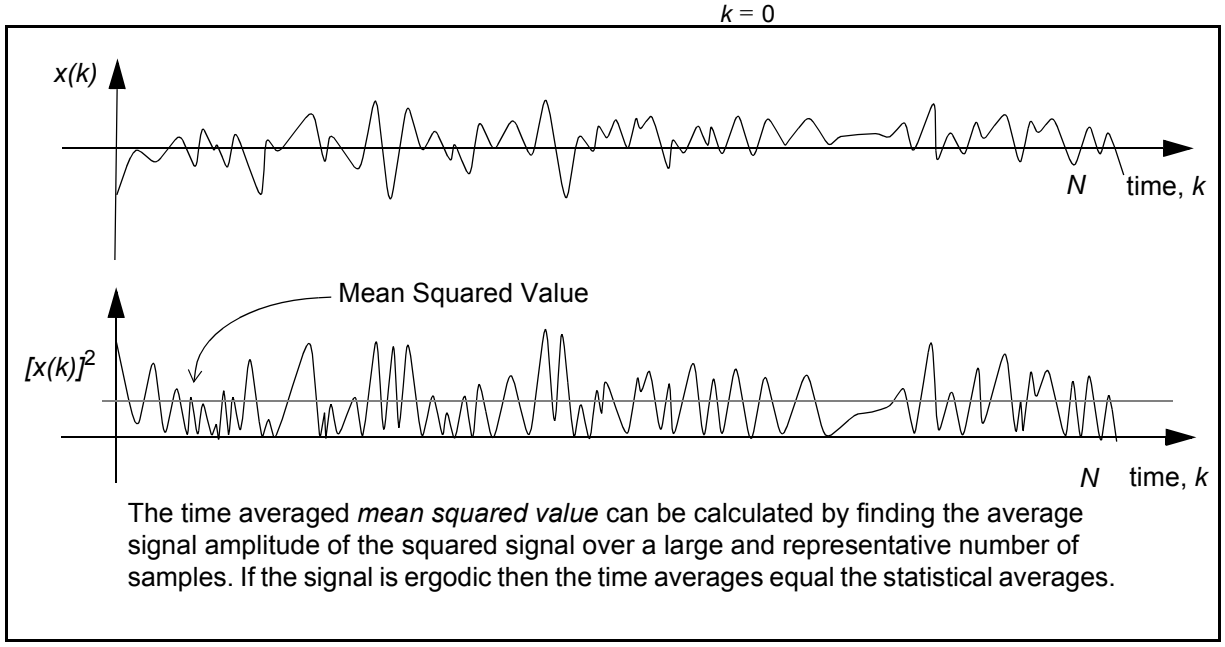
If the signal is ergodic then the time averages and statistical averages are the same. See also *Ergodic, Expected Value, Mean Squared Value, Wide Sense Stationarity*.

Mean Squared Value: The statistical *mean squared value* of a signal, $x(k)$, is the average amplitude of the signal. Statistical *mean squared value* is often denoted using the statistical expectation operator, $E\{.\}$, which is calculated as:

$$E\{x^2(k)\} = \text{Statistical Mean Squared Value of } x(k) = \sum_{k} x^2(k)p\{x(k)\} \tag{426}$$

where $p\{x(k)\}$ is the probability density function of $x(k)$. In real time DSP the probability density function of a signal is rarely known and therefore to find the *mean squared value* of a signal then the more intuitively obvious calculation of a time average calculated over a large and representative number of samples, *N*, is used:

$$\text{Average Squared Value} = \frac{1}{N}\sum_{k=0}^{N-1} x^2(k) \; . \tag{427}$$



The time averaged *mean squared value* can be calculated by finding the average signal amplitude of the squared signal over a large and representative number of samples. If the signal is ergodic then the time averages equal the statistical averages.

If the signal is ergodic then the time averages and statistical averages are the same. Note that *mean squared value* is always a positive value for any non-zero signal. See also *Ergodic, Expected Value, Mean Squared Value, Variance, Wide Sense Stationarity*.

Memory: Integrated circuits used to store binary data. Most *memory* devices are CMOS semiconductors. For a DSP system *memory* will either be ROM or RAM. See also *Static RAM, Dynamic RAM*.

Message: The information to be communicated in a communication system. The *message* can be continuous (analog) or discrete (digital). If an analog *message* is to be transmitted via a digital communications system it must first be sampled and digitized. See also *Analog to Digital Converter, Digital Communications.*

MFLOPS: This measure gives the *speed* rating of processor in terms of the number of millions of floating point operations per second (MFLOPS) a processor can do. DSP processors can often perform more FLOPS than their clock speeds. This counter-intuitive capacity results from the fact that the floating point operations are pipeline -- with MFLOPS calculated as a time-averaged (best case) performance. The MFLOPS rating can be misleading for practical programs running on a DSP processor that rarely attain the MFLOPS speed when performing peripheral functions such as data acquisition, data output, etc.

Middle A: See *Western Music Scale.*

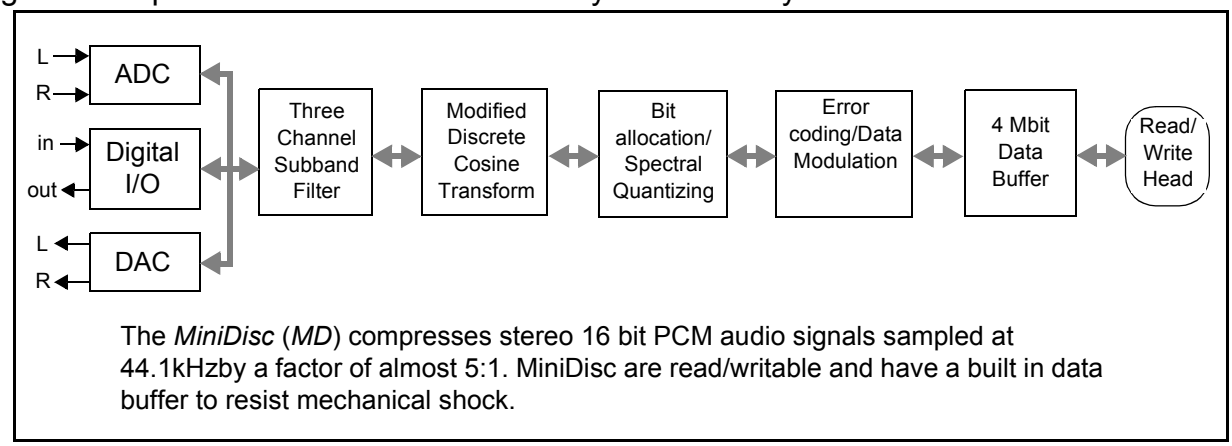
Middle C: See *Western Music Scale.*

MiniDisc (MD): The *MiniDisc* was introduced to the audio market in 1992 as a digital audio playback and record format with the aim of competing with both compact disc (CD) introduced in 1983, and the compact cassette introduced in the 1960s. Sony developed the *MiniDisc* partly to break into the portable hifidelity audio market and therefore the format need to be compact and resistant to vibration and mechanical knocks [155]. Compared to the very successful CD format, the

MiniDisc offers the advantage of being much smaller by virtue of smaller media requited by psychoacoustically compressed data. In addition, it features a record facility. The *MiniDisc* is a competing format to Philip's DCC which also uses psychoacoustic data compression techniques.

The *MiniDisc* is 64mm in diameter and uses magneto-optical techniques for recording. The size of the disc was kept small by using adaptive transform acoustic coding (ATRAC) to compress original 44.1kHz, 16 bit PCM music by a factor of 4.83. One *MiniDisc* can store 64 minutes of compressed stereo audio requiring around 140 Mbytes. Space is also made available for timing and track information. . The *MiniDisc* encodes data using the same modulation and similar error checking as the CD, namely eight to fourteen modulation (EFM) and a slightly modified cross interleaved Reed-Solomon coding (CIRC).

The risk of shock and vibration in everyday use is addressed by a 4Mbit buffer capable of storing more than 14 seconds of compressed audio. Therefore if the optical pickup loses its tracking the music can continue playing while the tracking is repositioned (requiring less than a second) and the buffer is refilled. In fact the pickup can read 5 times faster that the ATRAC decoder and therefore during normal operation the *MiniDisc* reads only intermittently.



The *MiniDisc* (*MD*) compresses stereo 16 bit PCM audio signals sampled at 44.1kHzby a factor of almost 5:1. MiniDisc are read/writable and have a built in data buffer to resist mechanical shock.

The *MiniDisc* can also be used for data storage and corresponds to a read-write disc of storage capacity 140Mbyte. See also *Adaptive Transform Acoustic Coding, Compact Disc (CD), Digital Audio, Digital Audio Tape (DAT), Digital Compact Cassette (DCC), Psychoacoustics*.

Minimum Audible Field: A measure of the lowest level of detectable sound by the human ear. See entry *Threshold of Hearing*.

Minimum Norm Vector: See *Vector Properties and Definitions - Minimum Norm.*

Minimum Phase: All zeroes of the transfer function lie within the unit circle on the z-plane. See also *Z-transform.*

Minimum Residual: See *Least Squares Residual*.

Minimum Shift Keying (MSK): A form of frequency shift keying in which memory is introduced from symbol to symbol to ensure continuous phase. The separation in frequency between symbols is *1/(2T)* Hz (for a symbol period of *T* seconds) allowing the maximum number of orthogonal signals in a fixed bandwidth. The fact that the MSK symbol stream is constrained to ensure continuous phase and has signals closely spaced in frequency means that MSK modulation is the most spectrally efficient form of FSK. MSK is sometimes referred to as Fast FSK since more data can be transmitted over a fixed bandwidth with MSK than FSK. Gaussian MSK (GMSK, as used in the GSM mobile radio system, for example) introduces a Gaussian pulse shaping on the MSK signals. This

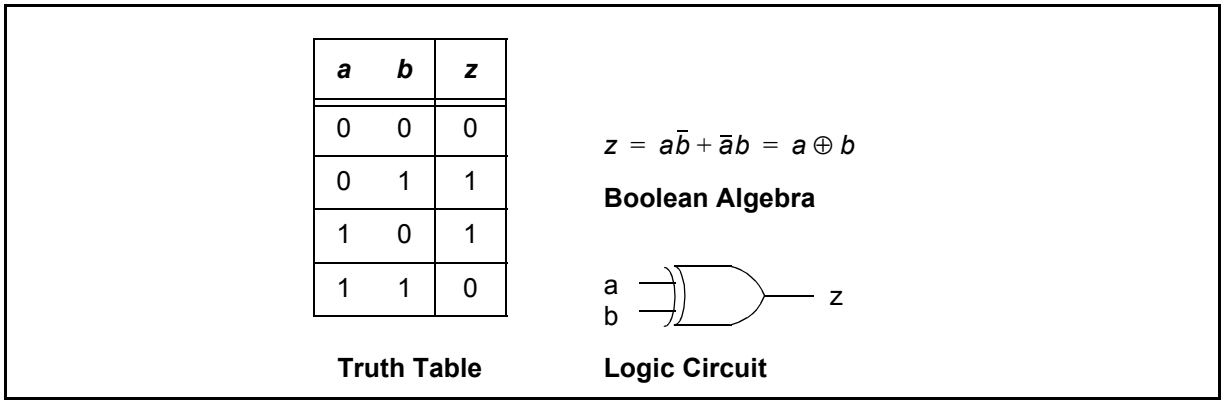
pulse shaping allows a trade-off between spectral overlap and interpulse interference. See also *Frequency Shift Keying, Continuous Phase Modulation.*

MIPS: This gives a measure of the number of MIPS (millions of instructions per second) that a DSP processor can do.

Modem: A concatenation of MODulate and DEModulate. *Modems* are devices installed at both ends of an analog communication line (such as a telephone line). At the transmitting end digital signals are modulated onto the analog line, and at the receiving end the incoming signal is demodulated back to digital format. *Modems* are widely used for inter-computer connection and on FAX machines.

Modular Interface eXtension (MIX): *MIX* is a high performance bus to connect expansion modules to a VME bus or a Multibus II baseboard. A few companies have adopted this standard.

Modulo-2 Adder: Another name for an exclusive OR gate. See also *Full Adder, Pseudo-Random Binary Sequence.*



a	*b*	*z*
0	0	0
0	1	1
1	0	1
1	1	0

Truth Table

$z = a\bar{b} + \bar{a}b = a \oplus b$

Boolean Algebra

Logic Circuit

Monaural: This refers to a system that presents signals to only one ear (e.g. a hearing aid worn on only one ear is monaural.) See also *Binaural, Monophonic, Stereophonic*.

Monaural Beats: When two tones with slightly different frequencies are played together, the ear may perceive a composite tone beating at the rate of the frequency difference between the tones. See also *Beat Frequencies, Binaural Beats.*

Monophonic: This refers to a system that has only one audio channel (although this single signal may be presented on multiple speakers). See also *Monaural, Stereophonic, Binaural.*

Moore-Penrose Inverse: See *Matrix Properties - Pseudo-Inverse.*

Mosaic: A hypertext browser used on the internet for interchange and exchange of information in the form of text, graphics, and audio. See also *Internet, World Wide Web.*

Most Significant Bit (MSB): The bit in a binary number with the largest arithmetic significance. See also *Least Significant Bit, Sign Bit.*

MSB LSB

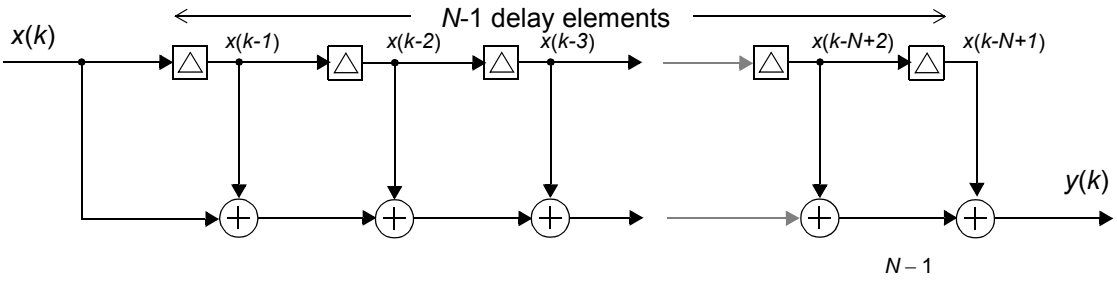
-128	64	32	16	8	4	2	1
0	0	0	1	1	0	1	1

$= 16 + 8 + 2 + 1 = 27_{10}$

Motherboard: A DSP board that has its own functionality, and also spaces for smaller functional boards (extra processors, I/O channels) to be inserted is called a motherboard. This is analogous to the main board on a PC system that is home to the processor and other key system components.

Moving Average (MA) FIR Filter: The moving average (MA) filter "usually" refers to an FIR filter of length *N* where all filter weights have the value of 1. (The term MA is however sometimes used to mean any (non-recursive) FIR filter usually within the context of stochastic signal modelling [77]).

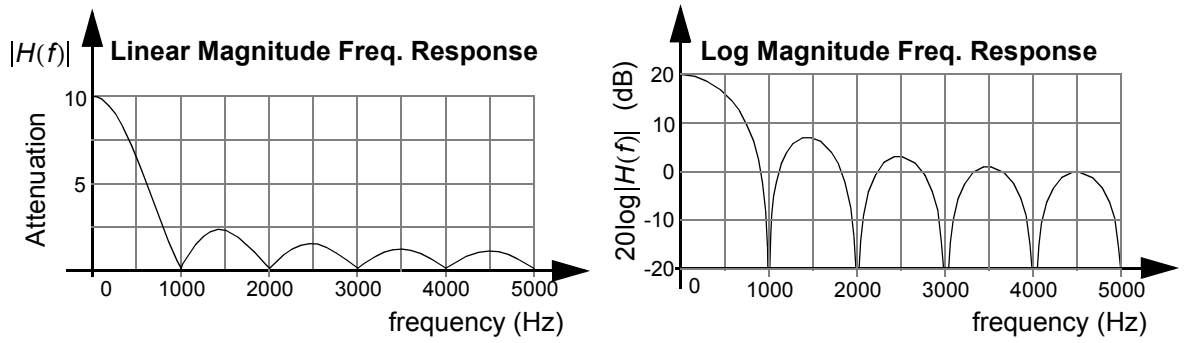
The moving average filter is a very simple form of low pass filter often found in applications where computational requirements need to be kept to a minimum. A moving average filter produces an output sample at time, *k*, by adding together the last N input samples (including the current one). This can be represented on a simple signal flow graph and with discrete equations as:



$$y(k) = x(k) + x(k-1) + x(k-2) + x(k-3) + ... + x(k-N+1) = \sum_{n=0}^{N-1} x(k-n)$$

The signal flow graph and output equation for a moving average FIR filter. The moving average filter requires no multiplications, only *N* additions.

As an example the magnitude frequency domain representations of a moving average filter with 10 weights is:



The linear and logarithmic frequency responses of a 10 weight moving average FIR filter. The peak of the first sidelobe of any moving average filter is always approximately 13dB below the gain at 0 Hz.

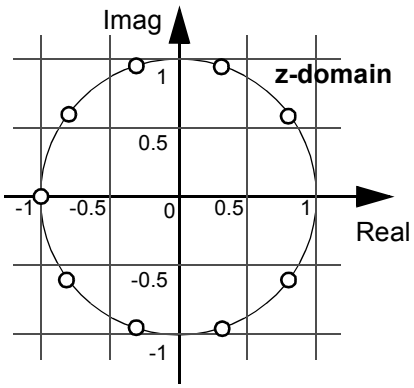
In terms of the z-domain, we can write the transfer function of the moving average FIR filter as:

$$\begin{aligned} H(z) = \frac{Y(z)}{X(z)} &= 1 + z^{-1} + z^{-2} + \ldots + z^{-N+1} \\ &= \sum_{i=0}^{N-1} z^{-i} = \frac{1 - z^{-N}}{1 - z^{-1}} \end{aligned} \tag{428}$$

recalling that the sum of a geometric series $\{1, r, r^2, \ldots, r^m\}$ is given by $(1 - r^{m+1})/(1 - r)$. If the above moving average transfer function polynomial is factorized, this therefore represents a transfer function with N zeroes and a single pole at $z = 1$, which is of course cancelled out by a zero at $z = 1$ since an FIR filter has no poles associated with it. We can find the zeroes of the polynomial in Eq. 428 by solving:

$$\begin{aligned} & 1 - z^{-N} = 0 \\ \Rightarrow z_n &= \sqrt[N]{1} \quad \text{where } n = 0 \ldots N - \\ \Rightarrow z_n &= \sqrt[N]{e^{j2\pi n}} \quad \text{noting } e^{j2\pi n} = \cdot \\ \Rightarrow z_n &= e^{\frac{j2\pi n}{N}} \end{aligned} \tag{429}$$

which represents N zeroes equally spaced around the unit circle starting at $z = 1$, but with the $z = 1$ zero cancelled out by the pole at $z = 1$. The pole-zero z-domain plot for the above 10 weight moving average FIR filter is:



$$H(z) = \frac{Y(z)}{X(z)} = 1 + z^{-1} + z^{-2} + \ldots + z^{-9} = \sum_{i=0}^{9} z^{-i} = \frac{1 - z^{-10}}{1 - z^{-1}}$$

The pole-zero plot for a moving average filter of length 10. As expected the filter has 9 zeroes equally spaced around the unit circle (save the one not present at $z = 1$). In some representations a pole and a zero may be shown at $z = 1$, however these cancel each other out. The use of a pole is only to simplify the z-transform polynomial expression.

In general if a moving average filter has N weights then the width of the first (half) lobe of the mainlobe is $f_s/2N$ Hz, which is also the bandwidth of all of the sidelobes up to $f_s/2$.

The moving average filter shown will amplify an input signal by a factor of N. If no gain (or gain =1) is required at 0 Hz then the output of the filter should be divided by N. However one of the attractive features of a moving average filter is that it is simple to implement and the inclusion of a division is not conducive to this aim. Therefore should 0 dB be required at 0 Hz, then if the filter length is made a power of 2 (i.e. 8, 16, 32 and so on) then the division can be done with a simple shift right operation of the filter output, whereby each shift right divides by 2.

The moving average FIR filter is linear phase and has a group delay equal to half of the filter length ($N/2$). See also *Comb Filter, Digital Filter, Exponential Averaging, Finite Impulse Response Filter, Finite Impulse Response Filters-Linear Phase, Infinite Impulse Response Filter*.

Moving Picture Experts Group (MPEG): The MPEG standard comes from the International Organization for Standards (ISO) sub-committee (SC) 29 which is responsible for standards on “Coding of Audio, Picture, Multimedia and Hypermedia Information”. Working Group (WG) 11 (ISO JTC1/SC29/WG11) considered the problem of coding of multimedia and hypermedia information and produced the *MPEG* joint standards with the International Electrotechnical Commission (IEC):

- ISO/IEC 11 172: MPEG-1 (Moving Picture Coding up to 1.5 Mbit/s)

  Part 1: Systems
  Part 2: Video
  Part 3: Audio
  Part 4: Compliance Testing (CD)
  Part 5: Technical Report on Software for ISO/IEC 11 172

- ISO/IEC 13 818: MPEG-2 (Generic Moving Picture Coding)

  Part 1: Systems (CD)
  Part 2: Video (CD)
  Part 3: Audio (CD)
  Part 4: Compliance Testing

Part 5: Technical Report on Software for ISO/IEC 13 818
Part 6: Systems Extensions
Part 7: Audio Extensions

Some current work of (ISO JTC1/SC29/WG11) is focussed on the definition of the *MPEG-4* standard for Very-low Bitrate Audio-Visual Coding.

MPEG-1 essentially defines a bit stream representation for the synchronized digital video and audio compressed to fit in a bandwidth of about 1.5Mbits/s, which corresponds to the bit rate output of a CD-ROM or DAT. The video stream requires about 1.15 Mbits/s, with the remaining bandwidth used by the audio and system data streams. *MPEG* is also widely used on the Internet as a means for transferring audio/video clips. *MPEG-1* has subsequently enabled the development of various multimedia systems and CD-DV (compact disc digital video).

The *MPEG* standard is aimed at using intra-frame (as in JPEG) and inter-frame compression techniques to reduce the digital storage requirement of moving pictures, or video [72]. *MPEG-1* video reduces the color subsampling ratio of a picture to one quarter of the original source values in order that actual compression algorithms are less processor intensive. *MPEG-1* video then uses a combination of the discrete cosine transform (DCT) and motion estimation to exploit the spatial and temporal redundancy present in video sequences and (depending on the resolution of the original sequence) can yield compression ratios of approximately 25:1 to give almost VHS quality video. The motion estimation algorithm efficiently searches blocks of pixels, and therefore can track the movement of objects between frames or as the camera pans around. The DCT exploits the physiology of the human eye by taking blocks of pixels and converting them from the spatial domain to the frequency domain with subsequent quantization. As with JPEG, a zig-zag scan of the DCT coefficients yields long runs of zero for the higher frequency components. This improves the efficiency of the run length encoding (also similar to JPEG).

In general very high levels of computing power are required for MPEG encoding (of the order of hundreds of MIPs to encode 25 frames/s. However decoding is not quite as demanding and there are a number of single chip decoder solutions available.

MPEG-2 is designed to offer higher than *MPEG-1* quality playback at bit rates of between 4 and 10Mbits/s which is above the playback rate currently achievable using CD disc technology . *MPEG-4* is aimed at very low bit rate coding for applications such as video-conferencing or video-telephony. See also *Compression, Discrete Cosine Transform, H-Series Recommendations - H261, International Organisation for Standards (ISO), Moving Picture Experts Group - Audio, Psychoacoustic Subband Coding, International Telecommunication Union, ITU-T Recommendations, Standards.*

Moving Picture Experts Group (MPEG) - Audio: The International Organization for Standards (ISO) *MPEG* audio standards were based around the developed compression techniques of MUSICAM (Masking Pattern Adapted Universal Subband Integrated Coding and Multiplexing) and ASPEC (Adaptive Spectral Perceptual Entropy Coding). *MPEG* audio compression uses subband coding techniques with dynamic bit allocation based on psychoacoustic models of the human ear. By exploiting both spectral and temporal masking effects, compression ratios of up to 12:1 for CD quality audio (without too much degradation to the average listener) can be realized.

The so called *MPEG-1*, ISO 11172-3 standard, describes compression coding schemes of hifidelity audio signals sampled at 48kHz, 44.1 kHz or 32 kHz with 16 bits resolution in one of four modes: (1) single channel; (2) dual (independent or bilingual) channels; (3) stereo channels; and (4) joint stereo . The standard only defines the format of the encoded data and therefore if improved

psychoacoustic models can be found then they can be incorporated into the compression scheme. Note that the psychoacoustic modelling is only required in the coder, and in the decoder the only requirement is to “unpack” the signals. Therefore the cost of an MPEG decoder is lower than an MPEG encoder.

The standard defines layers 1, 2 and 3 which correspond to different compression rates which require different levels of coding complexity, and of course have different levels of perceived quality. The various parameters (based on an input signal sampled at 48 kHz with 16 bits samples - a data rate of 768 kbits/s) of the three layers of the model are:

MPEG Audio ISO 11172-3 Standard	Theoretical coding/ decoding delay (ms)	Target bit rate/channel (kbits/s)	Comp-ression ratio	No of subbands in psycho-acoustic model	“Similar” compression schemes
Layer 1	19	192	4:1	32	PASC
Layer 2	35	128	6:1	32	MUSICAM
Layer 3	59	64	12:1	576	ASPEC

Layer 1 is the least complex to implement and is suitable for applications where good quality is required and audio transmission bandwidths of at least 192 kbits/s are available. PASC (precision adaptive subband coding) as used on the digital compact cassette (DCC) developed by Philips is very similar to layer 1. Layer 2 is identical to MUSICAM. Layer 3 which achieves the highest rate of data compression is only required when bandwidth is seriously limited; at 64 kbits/s the quality is generally good, however a keen listener will notice artifacts.

In the MPEG-2, ISO 13818-3 standard, key advancements have been made over MPEG-1 ISO 11172 with respect to inclusion of dynamic range controls, surround sound, and the use of lower sampling rates. Surround sound, or multichannel sound is likely to be required for HDTV (high definition television) and other forms of digital audio broadcasting. Draft standards for multichannel sound formats have already been published by the International Telecommunication Union - Radiocommunication Committee (ITU-R) and European Broadcast Union (EBU). MPEG-2 is designed to transmit 5 channels, 3 front channels and 2 surround channels in so called 3/2 surround format. Using a form of joint stereo coding the bit rate for layer 2 of MPEG-2 will be about 2.5 times the 2 channel MPEG-1 layer 2, i.e. between 256 and 384 bits/sec.

MPEG-2 was also aimed at extending psychoacoustic compression techniques to lower sampling frequencies such as (24 kHz, 22.05 kHz and 16 kHz) which will give good fidelity for speech only type tracks. It is likely that this type of coding could replace techniques such as the ITU-T G.722 coding (G - series recommendations).

MPEG-4 will code audio at very low bit rates and is currently under consideration. See also *Psychoacoustics, Precision Adaptive Subband Coding (PASC), Spectral Masking, Temporal Masking*

MPEG: See *Moving Picture Experts Group*.

Multichannel LMS: See *Least Mean Squares Algorithm Variants.*

Multimedia: The integration of speech, audio, video and data communications on a computer. For all of these aspects DSP co-processing may be necessary to implement the required computational

algorithms. Multimedia PCs have integrated FAX, videophone, audio and TV - all made possible by DSP.

Multimedia and Hypermedia Information Coding Experts Group (MHEG): *MHEG* is a standard for hypermedia document representation. *MHEG* is useful for the implementation aspects of interactive hypermedia applications such as on-line textbooks, encyclopedias, and learning software such as are already found on CD-ROM [94].

The *MHEG* standard comes from the International Organization for Standards (ISO) sub-committee (SC) 29 which is responsible for standards on “Coding of Audio, Picture, Multimedia and Hypermedia Information”. Working Group (WG) 12 (ISO JTC1/SC29/WG12) considered the problem of coding of multimedia and hypermedia information and produced the *MHEG* joint standard with the International Electrotechnical Commission (IEC): ISO/IEC 13522 *MHEG* (Coding of Multimedia and Hypermedia Information).

See also *International Organisation for Standards, Multimedia, Standards*.

Multimedia Standards: The emergence of multimedia systems in the 1990s brings the communication and presentation of audio, video, graphics and hypermedia documents onto a common platform. The successful integration of software and hardware from different manufacturers etc requires that standards are adopted. For current multimedia systems a number of ITU, ISO and ISO/IEC JTC standards are likely to be adopted. A non-exhaustive sample list of standards that are suitable include:

- ITU-T Recommendations:

  F.701 Teleconference service.

  F.710 General principles for audiographic conference service.

  F.711 Audiographic conference teleservice for ISDN.

  F.720 Videotelephony services - general.

  F.721 Videotelephony teleservice for ISDN.

  F.730 Videoconference service- general.

  F.732 Broadband Videoconference Services.

  F.740 Audiovisual interactive services.

  G.711 Pulse code modulation (PCM) of voice frequencies.

  G.712 Transmission performance characteristics of pulse code modulation.

  G.720 Characterization of low-rate digital voice coder performance with non-voice signals.

  G.722 7 kHz audio-coding within 64 kbit/s; Annex A: Testing signal-to-total distortion ratio for kHz audio-codecs at 64 kbit/s.

  G.724 Characteristics of a 48-channel low bit rate encoding primary multiplex operating at 1544 kbit/s.

  G.725 System aspects for the use of the 7 kHz audio codec within 64 kbit/s.

  G.726 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM). Annex A: Extensions of Recommendation G.726 for use with uniform-quantized input and output.

  G.727 5-, 4-, 3- and 2-bits sample embedded adaptive differential pulse code modulation (ADPCM).

  G.728 Coding of speech at 16 kbit/s using low-delay code excited linear prediction. Annex G to Coding of speech at 16 kbit/s using low-delay code excited linear prediction: 16 kbit/s fixed point specification.

H.221 Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices

H.242 System for establishing communication between audiovisual terminals using digital channels up to 2 Mbit/s.

H.261 Video codec for audiovisual services at p x 64 kbit/s.

H.320 Narrow-band visual telephone systems and terminal equipment.

T.80 Common components for image compression and communication - basic principles.

X.400 Message handing system and service overview (same as F.400).

- Proprietary Standards:

Bento Sponsored by Apple Inc for multimedia data storage.

GIF Compuserve Inc graphic interchange file format.

QuickTime Digital video replay on the MacIntosh.

RIFF Microsoft and IBM multimedia file format.

DVI Intel's digital video.

MIDI Musical digital interface.

- International Organization for Standards:

HyTime Hypermedia time based structuring language.

IIF Image interchange format.

JBIG Lossless compression for black and white images.

JPEG Lossy compression for continuous tone, natural scene images.

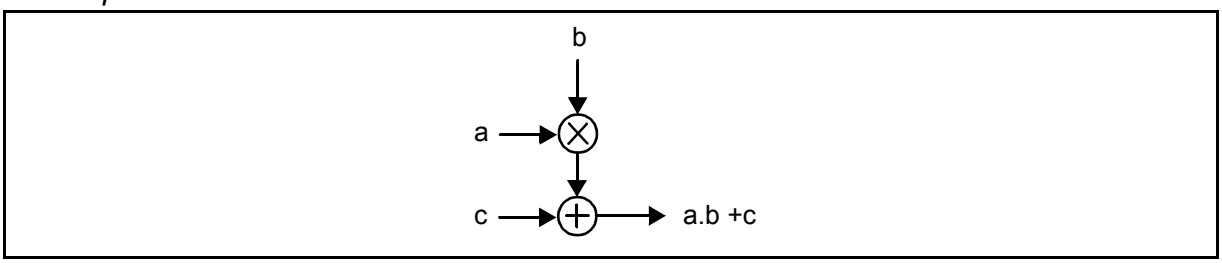
MHEG Multimedia and hypermedia information coding.

MPEG Digital video compression techniques.

ODA Open document architecture.

See also *International Telecommunication Union, International Organisation for Standards, Standards*.

Multiply Accumulate (MAC): The operation of multiplying two numbers and adding to another value, i.e. $((a \times b) + c)$. Many DSP processors can perform (on average) one MAC in one instruction cycle. Therefore if a DSP processor has a clock speed of 20MHz, then it can perform a peak rate of 20,000,000 *multiply and accumulates* per second. See also *DSP Processor, Parallel Adder, Parallel Multiplier.*



Multiprocessing: Using more than one DSP processor to solve a particular problem. The TMS320C40 has six I/O ports to communicate with other TMS320C40s with independent DMA. The term *multiprocessing* is sometimes used interchangeably with parallel processing.

Multipulse Excited Linear Predictive Coding (MLPC): *MLPC* is an extension of LPC for speech compression that goes some way to overcoming the false synthesized sound of LPC speech.

Multipurpose Internet Mail Extensions (MIME): *MIME* is a proposed standard from the Internet Architecture Board and supports several predefined types of non-text (non-ASCII) message contents, such as 8 bit 8kHz sampled μ-law encoded audio, GIF image files, and postscript as well as other forms of user definable types. See also *Standards*.

Multirate: A DSP system which performs computations on signals at more than one sampling rate usually to achieve a more efficient computational schedule. The important steps in a *multirate* system are decimation (reducing the sampling rate), and interpolation (increasing the sampling rate). Sub-band systems can be described as *multirate*. See also *Decimation, Interpolation, Upsampling, Downsampling, Fractional Sampling Rate Conversion*.

μ-law: Speech signals, for example, have a very wide dynamic range: Harsh “oh” and “b” type sounds have a large amplitude, whereas softer sounds such as “sh” have small amplitudes. If a uniform quantization scheme were used then although the loud sounds would be represented adequately the quieter sounds may fall below the threshold of the LSB and therefore be quantized to zero and the information lost. Therefore companding quantizers are used such that the quantization level at low input levels is much smaller than for higher level signals. Two schemes are widely in use: the μ-law in the USA and the A-law in Europe. The expression for μ-law compression is given by:

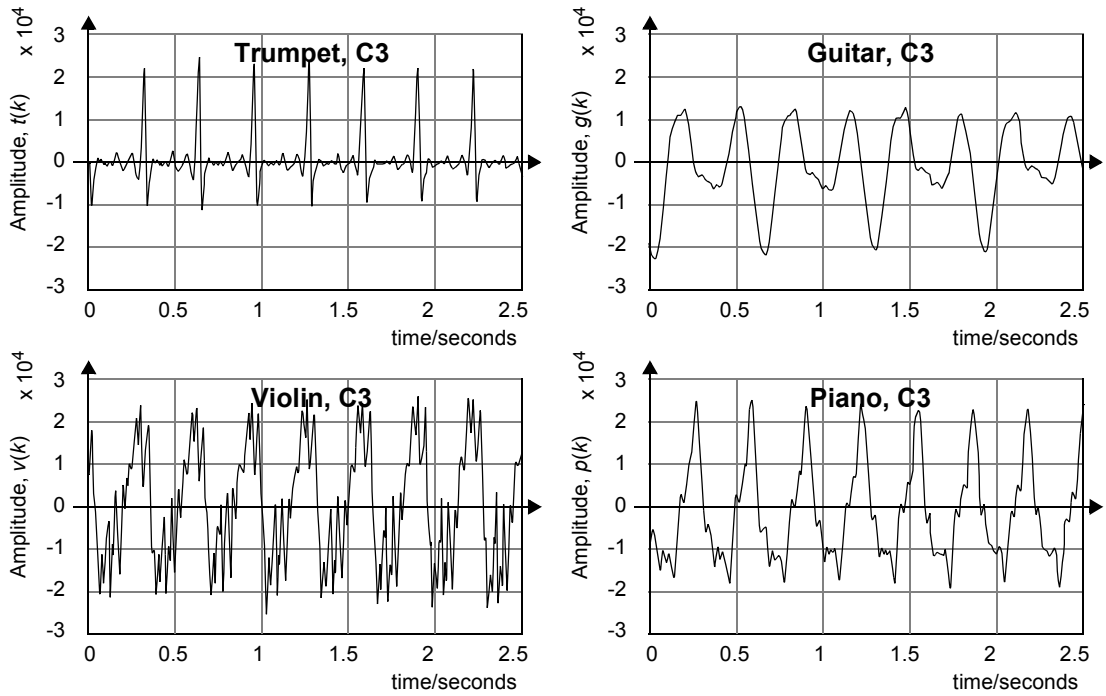
$$y(x) = \frac{ln(1+\mu x)}{ln(1+\mu)} \tag{430}$$

with *y(x)* being the compressed output for input *x*, and the function being negative symmetric around *x=0*. A typical value of μ is 255. See also *A-Law*.

Music: *Music* is a collection of sounds arranged in an order that sounds cohesive and regular. Most importantly, the sound of *music* is pleasant to listen to. Music can have has two main elements: a quasi-periodic set of musical notes and a percussive set of regular timing beats. Each musical note or discrete sound in *music* is characterized by a fundamental frequency and a rich set of harmonics, whereas the percussion sounds are more random (although distinctive) in nature [13], [14].

Many different ordered music scales (sets of constituent notes) exist. The most familiar is the 12 notes in an octave of the Western music scale on which most modern and classical music is played. The fundamental frequency of each note on the Western music scale can be related to the fundamental frequency of all other notes by a simple ratio. The same musical notes on different musical instruments are characterized by the harmonic content and the volume envelope. The

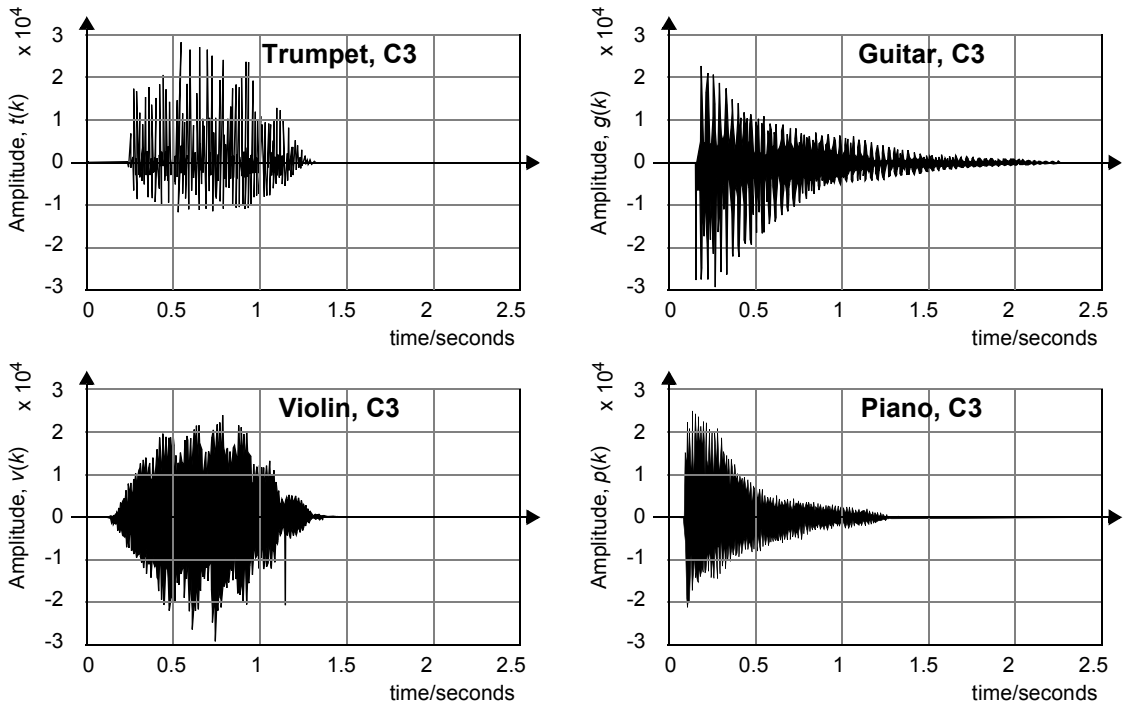
following figure shows the characteristic waveform for a sampled 0.03 second segment of C_4 note played on a trumpet, guitar, violin and piano:



Digitally sampled time waveforms representing the variation in sound pressure level of 0.03 second segments of a C_4 note (fundamental frequency of 261.6Hz on the Western music scale) played on a trumpet, guitar, violin and piano. The samples were taken from the full notes shown in the figures below.

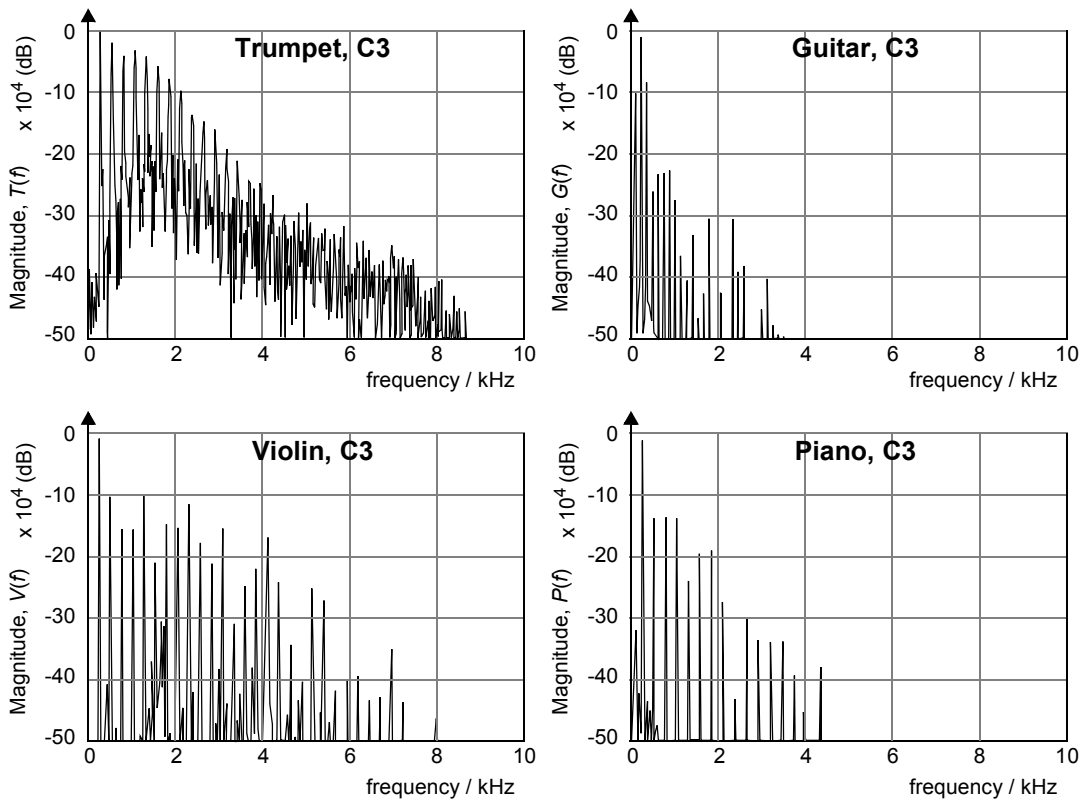
Clearly, although all of the instruments have a similar fundamental frequency, the varying harmonic content gives them completely different appearances in the time domain. The volume envelope of

a musical note also contributes to the characteristic sound, as shown in the following figure (from which the above 0.03 time segments were in fact taken):



Time waveforms showing the sound pressure level volume envelope of a C_3 note (fundamental frequency of 261.6Hz on the Western music scale) played on a trumpet, guitar, violin and piano. The amplitude envelope of the different musical instruments can be clearly seen.

To see the harmonic content of each of the four musical instruments we can perform a 2048 point FFT on a representative portion of the waveform resulting in the following frequency domain plots:



Frequency spectra of a C_4 note (fundamental frequency of 261.6Hz on the Western music scale) for a trumpet, guitar, violin and piano. The spectra were generated from a 0.05 second segment of the note.

Musical instruments are carefully designed to give them flexible tuning capabilities and, where possible, good natural frequency resonating. For example violins can be designed such that significant frequencies (such as A_4, of fundamental frequency 440Hz) corresponds to the resonance of the lower body of the instrument which as a result will enhance the sound, and also the feeling and tactile feedback to the violinist [14]. Clearly the subtleties of the generation and analysis of music is very complex, although the appreciation of music is very simple!

There are many other *music* scales such as the 22 note Hindu scale, and many other different Asian scales. This perhaps explains why when someone who has never experienced Chinese *music* listens to it for the first time it may be perceived off key and dissonant because it contains various notes that are just not present in the familiar Western *music* scale. Another example of an instrument that is not quite playing to the Western *music* scale are the Scottish bagpipes. The high notes on the chanter are not in fact a full octave (frequency ratio of 2:1) above the analogous lower notes. Hence the bagpipes can sound a little flat at the high notes. However, if the bagpipes are the sound to which we had become accustomed, and anything else might not sound right!

Music synthesis is now largely achieved using digital synthesizers that use a variety of DSP techniques to produce an output. See also *Digital Audio, Percussion, Music Synthesis, Sound Pressure Level, Western Music Scale*.

Music Synthesis: Most modern synthesizers use digital techniques to produce simulated musical instruments. Most synthesis requires setting up the fundamental frequency components with appropriate relative harmonic content and a suitable volume profile. A good overview of this area can be found in [14], [32]. See also *Attack-Decay-Sustain-Release, Granular Synthesis, LA Synthesis, Music*.

N

n: ”*n*” (along with “k” and “i”) is often used as a discrete time index for in DSP notation. See *Discrete Time*.

Narrowband: Signals are defined as *narrowband* if the fractional bandwidth of the signals is small, say <10%. See also *Fractional Bandwidth, Wideband*.

Nasals: One of the elementary sounds of speech, namely plosives, fricatives, sibilant fricative, semi-vowels, and nasals. Nasals are formed by lowering the soft palate of the mouth so blocking the mouth and forcing the air stream to pass out via the nose, as in the letter “m”. See also *Fricatives, Plosives, Semi-vowels, and Sibilant Fricatives*.

Natural Frequency: See *Resonant Frequency.*

Near End Echo: Signal echo that is produced by components in local telephone equipment. Near end echo arrives before far end echo. See also *Echo Cancellation, Far End Echo*.

Neper: The *neper* is a logarithmic measure used to express the attenuation or amplification of voltage or current where the natural logarithm (base $e = 2.71828...$) is used rather than the more normal base 10 logarithm:

$$\text{Neper (Np)} = \ln\left(\frac{V_{\text{out}}}{V_{\text{in}}}\right) \tag{431}$$

A decineper is calculated by multiplying the neper quantity by 10 (rather than 20 as would be used for decibels):

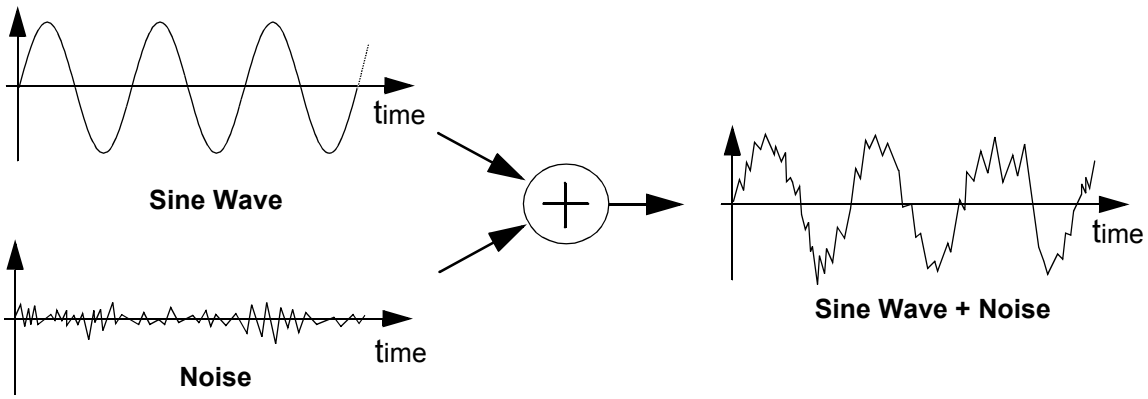
$$\text{Decineper (dNp)} = 10\ln\left(\frac{V_{\text{out}}}{V_{\text{in}}}\right) \tag{432}$$

To convert from *nepers* to decibels simply multiply by $20\log e = 8.686...$. The *neper* should not be confused with the Scottish word for turnips (or swedes) which is the neep. Traditionally neeps are eaten on 25th January each year to celebrate the birthday of Robert Burns, the Scottish poet who popularized Auld Lang Syne as well as many other of his own songs and poems. Neeps can of course be eaten at other times of the year. There is no known means by which neeps can be converted to decibels.

Neural Networks: Over the last few years the non-linear processing techniques known as *neural networks* have been used to solve a wide variety of DSP related problems such as speech recognition and image recognition [18], [112], [24]. The simplest forms of *neural network* can be directly related to the adaptive LMS filter, however the multi-layer nature of even these simple networks have very high computational loads. The name derives from the similarity of the computational model to a simplified model of the nervous system in animals. The applications and implementation of *neural networks* in DSP is set to grow in the next few years.

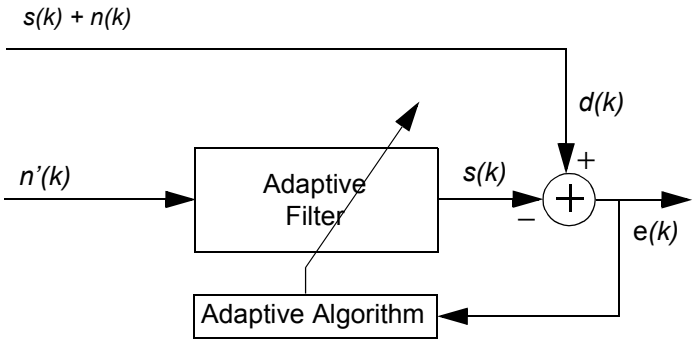
Newton LMS: See *Least Mean Squares Algorithm Variants.*

Noise: An unwanted component of a signal which interferes with the signal of interest. Most signals are contaminated by some form of noise, either present before sensing, or actually induced by the process of sensing the signal (conversion to electrical form) or the sampling process (quantization noise). Computations on a DSP processor can also induce various forms of arithmetic noise (round-off noise). Most DSP algorithms assume that noise sources can be well modelled as additive, i.e., the noise is *added* to the signal of interest. See also *Round-Off Noise, Truncation, White Noise, Additive White Gaussian Noise.*



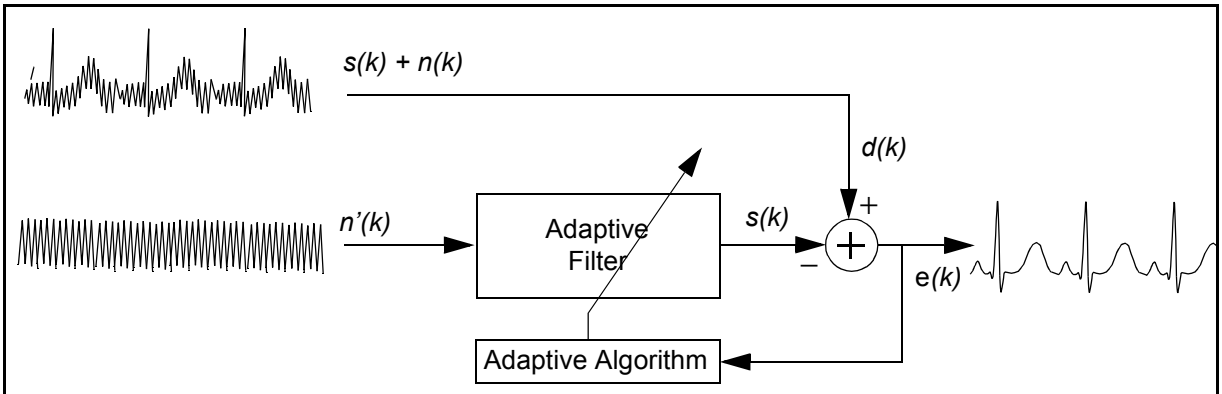
A Sine wave corrupted by additive noise.

Noise Cancellation: Using adaptive signal processing techniques, noise cancellation can be used to remove noise from a signal of interest in situations where a correlated reference of the noise signal is available:.

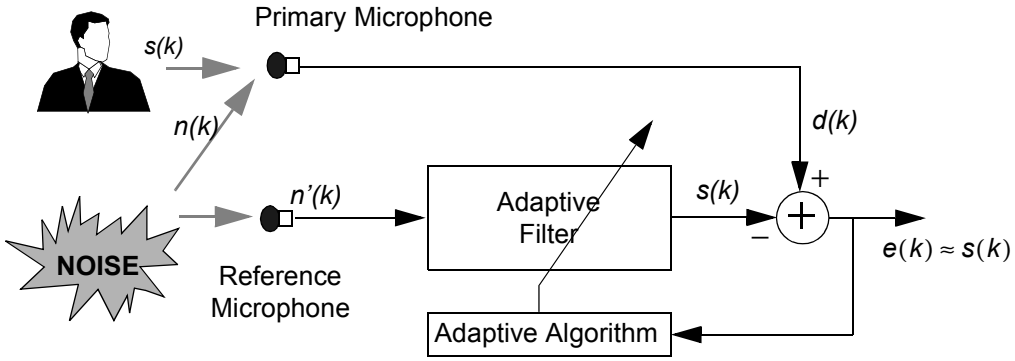


Generic adaptive signal processing noise canceller. Signal $s(k)$ is uncorrelated with $n(k)$ or $n'(k)$. However $n(k)$ and $n'(k)$. are correlated.

Noise cancellation techniques are found in biomedical applications where, for example it is required to remove mains hum periodic noise from an ECG waveform:



Adaptive noise cancellation of an ECG signal corrupted by mains hum.



Adaptive noise cancellation of a speech signal corrupted by noise. The reference microphone picks up the noise only, whereas the primary microphone picks both noise and speech. Note that if the reference microphone also picks up speech then the adaptive noise canceller will try to also cancel the speech signal. (This is clearly not the desired effect!)

See also *Active Noise Control, Adaptive Line Enhancer, Adaptive Filter, Echo Cancellation, Least Mean Squares Algorithm, Recursive Least Squares.*

Noise Control: See *Active Noise Control, Noise Cancellation.*

Noise Dosemeter: For persons subjected to noise at the workplace, a noise dosemeter or sound exposure meter can be worn which will average the "total" sound they are exposed to in a day. The measurements can then be compared with national safety standards [46].

Noise Shaping: A technique used for audio signal processing and sigma delta analog to digital converters where quantisation noise is high pass filtered out of the baseband. See also *Oversampling, Sigma Delta*.

Noncausal: See *Causal.*

Noncoherent: See *Coherent.*

Nonlinear: Not linear. See also *Linear System, Non-linear System.*

Non-linear System: A *non-linear system* is one that does **not** satisfy the linearity criteria such that if:

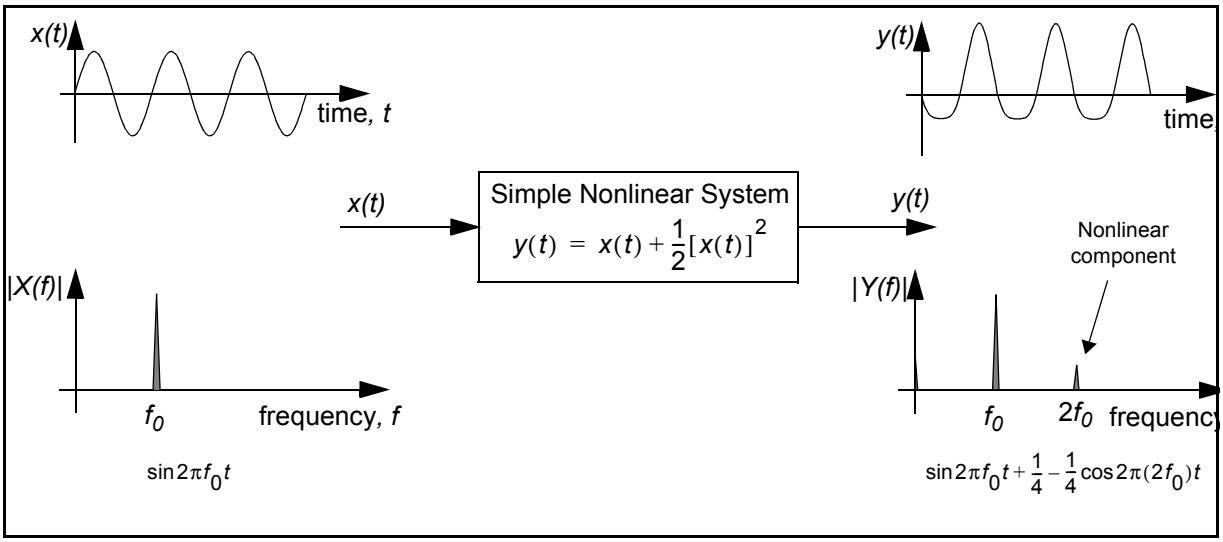
$$\begin{aligned} y_1(n) &= f[x_1(n)] \\ y_2(n) &= f[x_2(n)] \end{aligned} \tag{433}$$

then:

$$a_1 y_1(n) + a_2 y_2(n) = f[a_1 x_1(n) + a_2 x_2(n)] \tag{434}$$

For example the system $y(n) = 1.2x(n) + 3.4(x(n))^2$ is *nonlinear* as it does not satisfy the above linearity criteria. Any system which introduces harmonic distortion or signal clipping is non-linear. Non-linear systems can be extremely difficult to analyse both mathematically and practically. Low levels of nonlinear components that are relatively small in magnitude are often ignored in the analysis and simulation of systems.

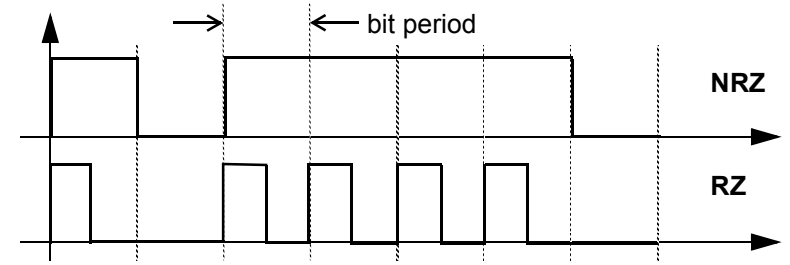
A simple way to test the linearity of a system is to input a single sine wave and vary the frequency over the bandwidth of interest and observe the output signal. If the output contains any sine wave components other than at the frequency of the input sine wave then it is *nonlinear* system. The most common form of nonlinearity is called harmonic distortion. See also *Distortion, Linear System, Total Harmonic Distortion, Volterra Filter.*



Non-negative Definite Matrix: See *Matrix Properties - Positive Definite.*

Non-Return to Zero (NRZ): When a stream of binary data is to be sent serially, such as transmission of PCM, the data can be sent as (half binary) return to zero (RZ), or (full binary) non-return to zero (NRZ). With RZ data streams after a 1 has been sent, the output waveform returns

back to 0, whereas with NRZ the output remains at 1 for the duration of the bit period. The waveform



The same sequence of bits, 1011110, transmitted as RZ and NRZ

assumed below is polar. See also *Bipolar (2), Polar.*

Non-Simultaneous Masking: See *Temporal Masking*.

Nonsingular Matrix: See *Matrix Properties - Nonsingular.*

Non-Volatile: Semiconductor memory that does not lose information when the power is removed is called *non-volatile*. ROM is an example of non-volatile memory. *Non-volatile* RAM is also available.

Norm: See *Vector Properties and Definitions - Norm.*

1-norm: See *Matrix Properties - 1-norm.*

2-norm: See *Matrix Properties - 2-norm.*

2-norm of a Vector: See *Vector Properties and Definitions - 2-norm.*

Normal Equations: In least squares error analysis the normal equation is given by:

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}_{LS} = \boldsymbol{A}^T\boldsymbol{b} \tag{435}$$

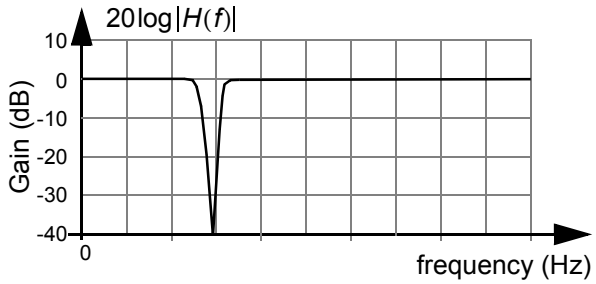
given the overdetermined system of equations:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{436}$$

where $\boldsymbol{A}$ is a known $m \times n$ matrix of rank n and with $m > n$, $\boldsymbol{b}$ is a known m element vector, and $\boldsymbol{x}$ is an unknown n element vector. See also *Least Squares, Overdetermined System, Underdetermined System*.

Normalised Step Size LMS: See *Least Mean Squares Algorithm Variants, Step Size Parameter.*

Notch Filter: A notch filter, $H(z)$ removes signal components at a very narrow band of frequencies:

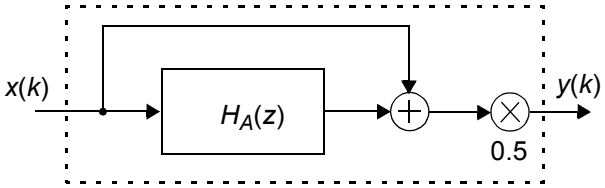


A notch filter removes a very narrow band of frequencies.

Notch filters can be designed using standard filter design techniques for band-stop filters. One form of notch filter can be designed using an all-pass IIR digital filter of the form:

$$H_A(z) = \frac{r^2 - 2r\cos\theta + z^{-2}}{1 - 2r\cos\theta + r^2 z^{-2}} \tag{437}$$

in the configuration:



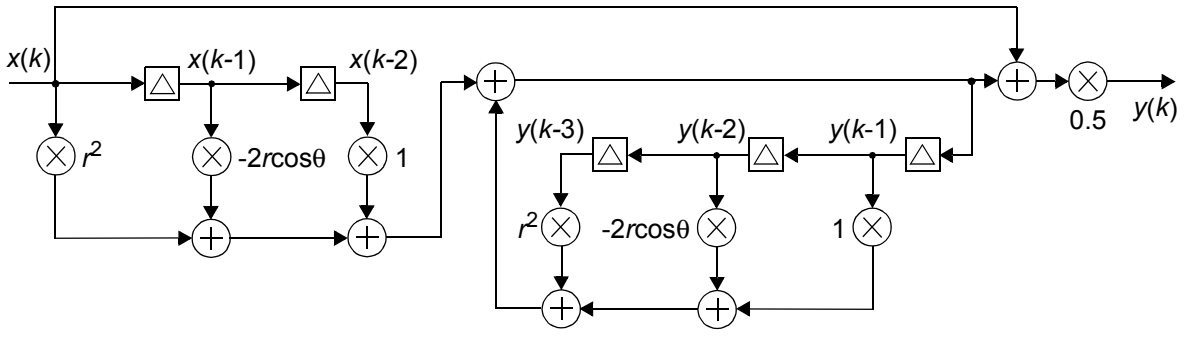
$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{2}(1 + H_A(z))$$

Notch filter designed using an all pass filter $H_A(z)$.

The parameters $\cos\theta$ and r are used to set the notch frequency and bandwidth of the notch. The notch frequency, f_n can be calculated from:

$$\cos\frac{2\pi f_n}{f_s} = \frac{2r\cos\theta}{1 + r^2} \tag{438}$$

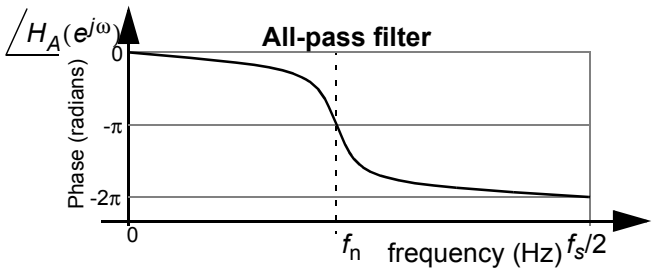
which is calculated from Eq. 437 by noting the frequency when the phase shift of the output of the all pass filter is $-\pi$ radians (see below). The above notch filter can be drawn more explicitly as the signal flow graph (SFG):



$$y(k) = r^2x(k) - 2r\cos\theta x(k-1) + x(k-2) + y(k-1) - 2r\cos\theta y(k-2) + r^2y(k-3)$$

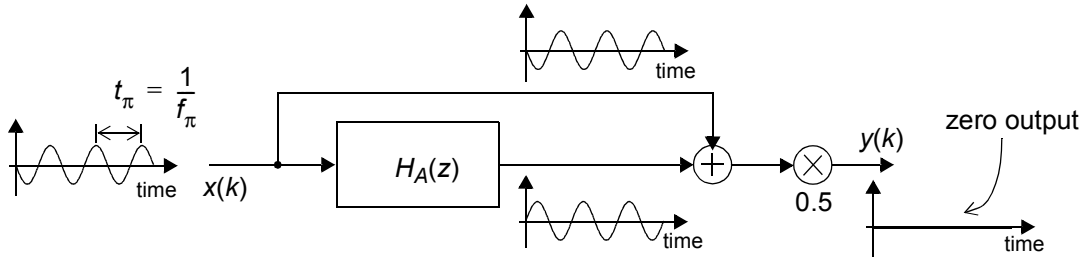
Signal flow graph for a notch filter based on an all-pass filter.

In order to appreciate the notch filtering attribute of this filter, note that the all pass filter $H_A(z)$ has a phase response of the form:



Typical form (i.e. -ve sigmoidal) phase response of the all-pass filter $H_A(z)$. The actual transition point through -π radians and the various graph slopes are determined by setting the parameters r and $\cos\theta$.

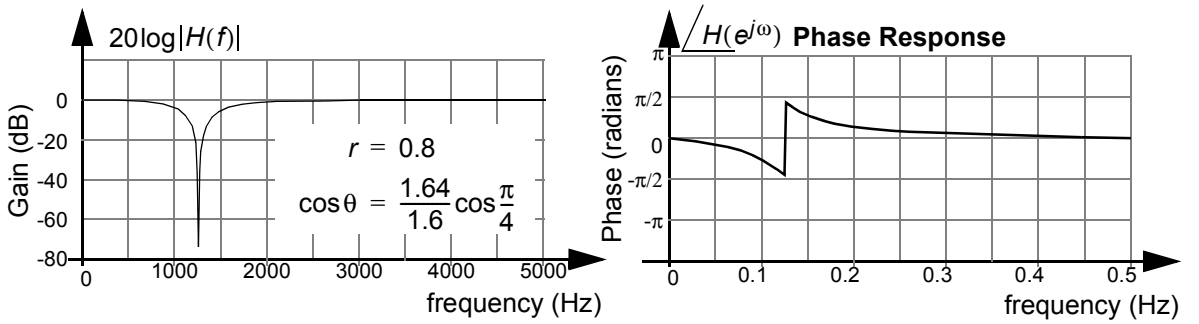
Therefore when the input signal is the frequency f_n, then the phase of the output signal of the all pass filter is exactly -π. When added to the input signal $x(k)$, the output $y(k)$ is zero:



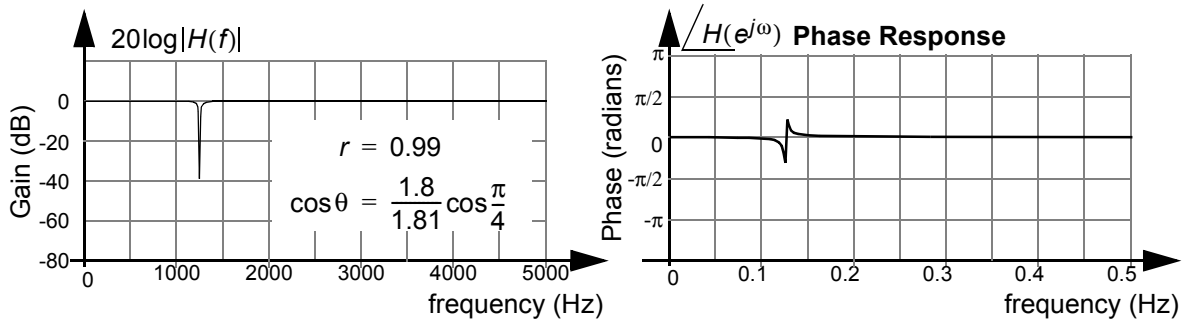
When the output of the all pass filter produces a phase shift of $-\pi$ radians for an input sine wave input of f_n Hz, the output, $y(k)$ of the notch filter is zero.

As examples, using Eq. 438 we can design two notch filters with a notch frequency of $f_n = 1250$ Hz, for a sampling rate of $f_s = 10000$ Hz. The first design has $r = 0.8$ and the second design has $r = 0.99$, thus giving different notch bandwidths :

Setting r close to 1 is equivalent to putting the poles and zeroes of the all-pass filter very close to the unit circle.



Notch filters at $f_n = 1250$ Hz, with $r = 0.8$ and $\cos\theta = (1.64/1.6)\cos\pi/4$.



Notch filter at $f_n = 1250$ Hz with $r = 0.99$ and $\cos\theta = (1.8/1.81)\cos\pi/4$. The notch bandwidth is smaller that the above design with $r = 0.8$ and $\cos\theta = (1.64/1.6)\cos\pi/4$. Note that the phase shift is very small at frequecies other than those near the notch frequency

If a notch filter is to be used to remove a "single" frequency, then adaptive noise cancellation can often be used as a suitable alternative if a suitable correlated noise source is available. See also *Adaptive Signal Processing, All-pass Filter, Digital Filter, Infinite Impulse Response Filter*.

Noy: The noy is a measurement of noisiness similar in its measurement to a phon. It is defined as the sound pressure level (SPL) of a band of noise from 910Hz to 1090 Hz that subjectively sounds as noisy as the sound under consideration [46]. See also *Equal Loudness Contours, Frequency Range of Hearing, Phons, Sound Pressure Level*.

Null Space: See *Vector Properties - Null Space*.

Numerical Integrity: Instability in a DSP system can either be (1) a function of feedback causing large unbounded outputs, or (2) when very large numbers are divided by very small numbers, or vice versa. Instability of type (2) can cause a loss of *numerical integrity* when the result is smaller than the smallest decimal number or larger than the largest decimal number that can be represented in the DSP processor being used. In the case of a number that is too small, then the result will likely be returned as zero. However if this number is to be used as a dividend the result is a divide by zero error, which will cause the algorithm to stop or become unstable by generating a maximum amplitude quotient.

As an example consider a particular microprocessor that has precision of 3 decimal places. The following matrix algorithm is to be implemented:

$$\boldsymbol{C} = [\boldsymbol{A}^{-1} + \boldsymbol{B}]^{-1} \tag{439}$$

Where,

$$\boldsymbol{A} = \begin{bmatrix} 1000 & 0 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} \tag{440}$$

Solving the problem using a processor with 3 decimal place of precision is straightforward and gives:

$$\begin{aligned} \boldsymbol{C} &= \left[\begin{bmatrix} 1000 & 0 \\ 0 & 1 \end{bmatrix}^{-1} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right]^{-1} \\ &= \left[\begin{bmatrix} 0.001 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right]^{-1} \\ &= \begin{bmatrix} 0.001 & 0 \\ 0 & 2 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1000 & 0 \\ 0 & 0.5 \end{bmatrix} \end{aligned} \tag{441}$$

However if the same problem was solved using a processor with only two places of decimal precision, then:

$$\begin{aligned} \boldsymbol{C} &= \left[\begin{bmatrix} 1000 & 0 \\ 0 & 1 \end{bmatrix}^{-1} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right]^{-1} \\ &= \left[\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right]^{-1} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}^{-1} \\ &= \text{non-invertible matrix} \end{aligned} \tag{442}$$

and the algorithm breaks down. See also *Ill-Conditioned*.

Numerical Properties: The ability of a DSP algorithm to produce intermediate results that are within the wordlength of the processor being used indicates that the particular algorithm has good *numerical properties*. If, for example, a particular DSP algorithm running on a 32 bit floating point DSP processor produces intermediate values that require more precision than 32 bits floating point, then clearly the final result will be in error by some margin. Therefore it is always desirable to used algorithms with good numerical properties. In linear algebra, for solving a linear set of equations the

QR algorithm is recognised as having good *numerical properties*, whereas Gaussian Elimination has very poor *numerical properties*. See also *Round-Off Noise*.

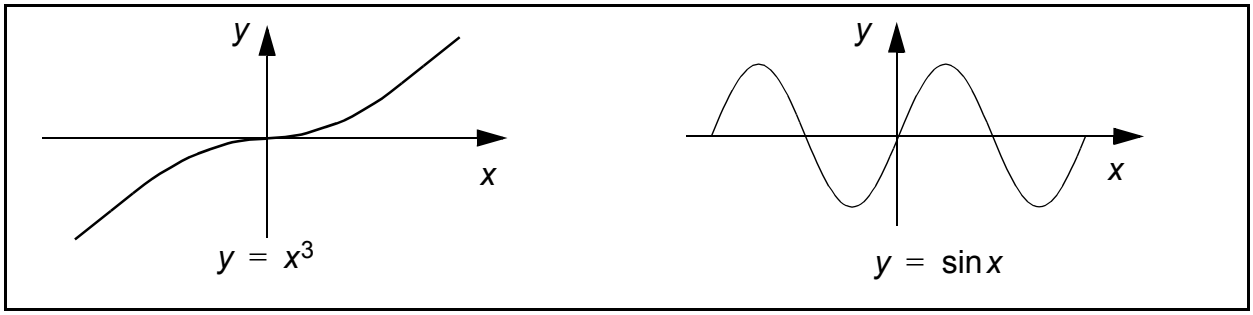
Numerical Stability: See *Numerical Integrity.*

Nyquist: The *Nyquist* frequency is the minimum frequency at which an analog signal must be sampled in order that no information is lost (assuming the sampling process is perfect). Mathematically, it can be shown that the *Nyquist* frequency must be greater than twice the highest frequency component of the signal being sampled in order to preserve all information [10]. In practical terms, real-world signals are never *exactly* bandlimited. However, the energy that gets aliased is kept small in properly designed DSP systems. See also *Aliasing.*

O

Octave: An *octave* refers the interval between two frequencies where one frequency is double to other. For example, from 125Hz to 250Hz is an *octave*, and from 250Hz to 500 Hz is an octave and so on. It may seem strange that *octave* derived from the Greek prefix "oct" which means eight, however this relates to the Western Music Scale whereby an *octave* is a set of eight musical notes (of increasing frequency), and where the first note has half of the frequency of the last note. See also *Decade, Logarithmic Frequency, Roll-off, Western Music Scale*.

Odd Function: The graph of an *odd function* has point symmetry about the origin such that $y = f(x) = -f(-x)$. For example both the functions $y = \sin x$ and $y = x^3$ are *odd functions*. In contrast an even function is symmetric about the y-axis such that $y = f(x) = f(x)$. See also *Even Function*.



Off-Line Processing: If recorded data is available on a hard disk and it is only required to process this data then store it back to disk then the computation is not time limited and this is referred to as *off-line processing*. If on the other hand an output must be generated as fast as an input is received from a real world sensor then this is *real-time processing*. See also *Real Time Processing*.

Offset Keyed Phase Shift Keying (OPSK or OKPSK): See *Offset Keying*.

Offset Keyed Quadrature Amplitude Modulation (OQAM or OKQAM): See *Offset Keying.*

Offset Keying: A modulation technique used with quadrature signals (i.e., those signals that can be described in terms of in-phase and quadrature, or cosine and sine, components). In *offset keying*, symbol transitions for the quadrature component are delayed one half a symbol period from those for the in-phase component.

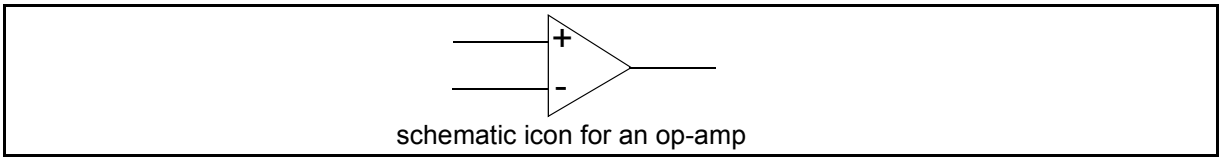
OnCE: Motorola on-chip emulator that allows easy debugging of the DSP56000 family of processors.

On-chip Memory: Most DSP processors (DSP56/96 series, TMS320, DSP16/32, ADSP 2100 etc.) have a few thousand words of *on-chip memory* which can be used for storing short programs, and (significantly) data. The advantage of *on-chip memory* is that it is faster to access than *off-chip memory*. For DSP applications such as a FIR filter, where very high speed is essential, the *on-chip memory* is very important. See also *DSP Processor, Cache.*

On-line Processing: See *Real Time Processing.*

Operational Amplifier (or Op-Amp): An integrated circuit differential amplifier that has a very high open-loop gain (of the order 100000), a high input impedance (MΩ), and low output impedance (100Ω) over a relatively small bandwidth. By introducing negative feedback around the amplifier,

gain ratios of 1-1000 over a wide bandwidth can be set up. *Op-Amps* are very widely used for many forms of signal conditioning in DSP audio, medical, telecommunication applications.



schematic icon for an op-amp

Oppenheim and Schafer: *Alan Oppenheim* and *Ronald Schafer* are the authors of the definitive 1975 text *Digital Signal Processing* published by Prentice Hall. Still a very relevant reference for DSP students and professionals, although since then many other excellent texts have been published.

Order of a Digital Filter: See *Digital Filter Order*.

Order Reversed Filter: See *Finite Impulse Response*.

Orthogonal Matrix: See *Matrix Properties - Orthogonal*.

Orthonormal Matrix: See entry for *Matrix Properties - Orthogonal*.

Orthogonal Vector: See *Vector Properties and Definitions - Orthogonal.*

Orthonormal Vector: See *Vector Properties and Definitions - Orthonormal.*

Otoacoustic Emissions: Sounds that are emitted spontaneously from the ear canal. Measurements of these emissions are used to diagnose hearing loss and other pathologies within the ear. The emissions are induced by stimulating the ear and then measured by recording the response produced after the stimulus.

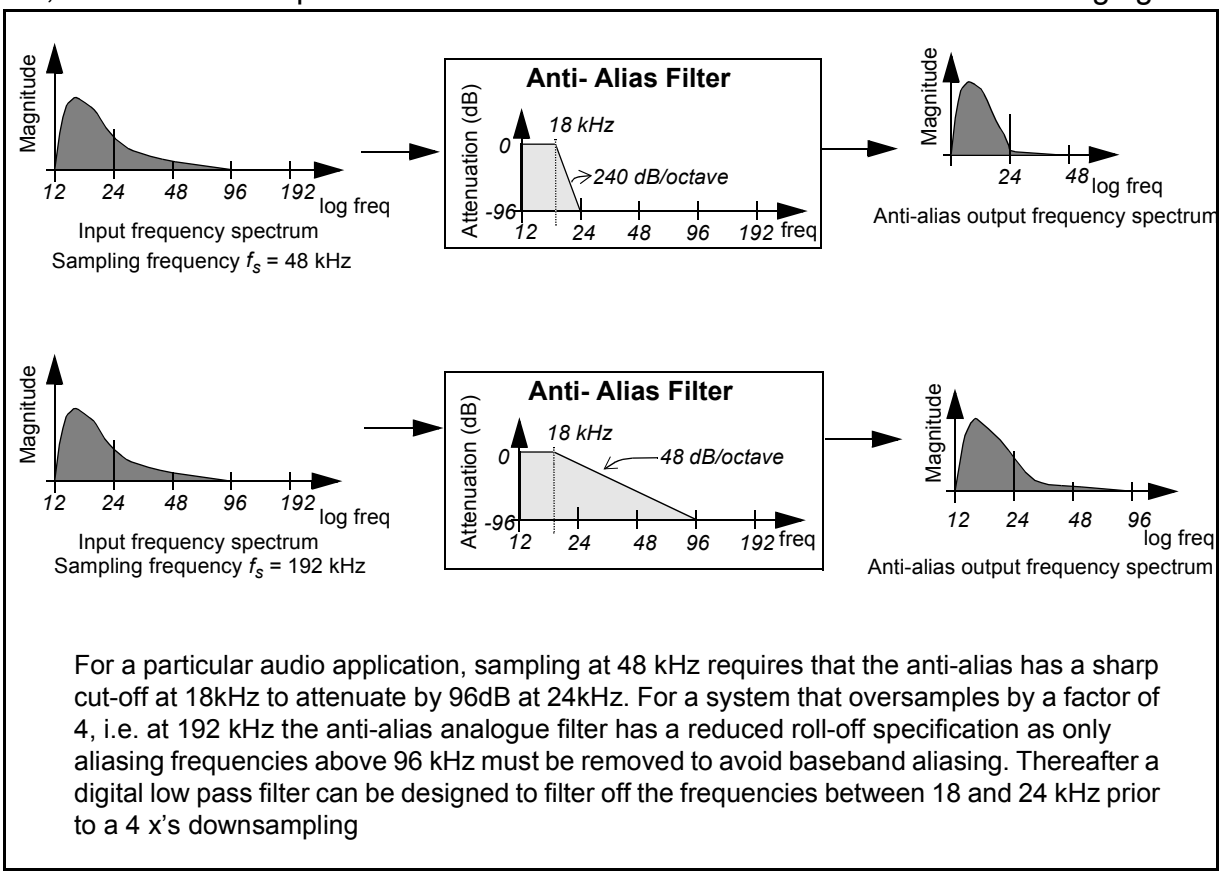
Outer Product: See *Vector Properties and Definitions - Outer Product.*

Overdetermined System of Equations: See *Matrix Properties - Overdetermined System of Equations.*

Oversampling: If a signal is sampled at a much higher rate than the Nyquist rate, then it is oversampled. *Oversampling* can bring two benefits: (1) a reduction in the complexity of the analog anti-alias filter; and (2) an increase in the resolution achievable from an *N*-bit ADC or DAC.

As an example of *oversampling* for reducing the complexity of the analog anti-alias filter, consider a particular digital audio system in which the sampling rate is 48kHz. The Nyquist criterion is satisfied by attenuating all frequencies above 24kHz that may be output by certain musical instruments (or interfering electronic equipment) by at least 96 dB (equivalent to a 16 bit dynamic

range). If it is decided that the low pass filter will cut off at 18 kHz, and if 96dB attenuation is required at 24kHz, then the filter requires a roll-off of 240 dB/octave as shown in the following figure:



For a particular audio application, sampling at 48 kHz requires that the anti-alias has a sharp cut-off at 18kHz to attenuate by 96dB at 24kHz. For a system that oversamples by a factor of 4, i.e. at 192 kHz the anti-alias analogue filter has a reduced roll-off specification as only aliasing frequencies above 96 kHz must be removed to avoid baseband aliasing. Thereafter a digital low pass filter can be designed to filter off the frequencies between 18 and 24 kHz prior to a 4 x's downsampling

Clearly this is a 40th order filter and somewhat difficult to reliably design in analogue circuitry! (Please note the figures used here are for example purposes only and do not necessarily reflect actual digital systems.) However if we *oversample* the music signal by 4 x's, i.e. at 4×48 kHz $=$ 192 kHz, then an analog anti-alias filter with a roll-off of only 48 dB/octave starting at 18 kHz and providing more than 96dB attenuation at half of the oversampled rate of 96 kHz is required as also shown in the above figure. (In actual fact the roll-off could be even lower as it is very unlikely there will be any significant frequency components above 30 kHz in the original analogue music.)

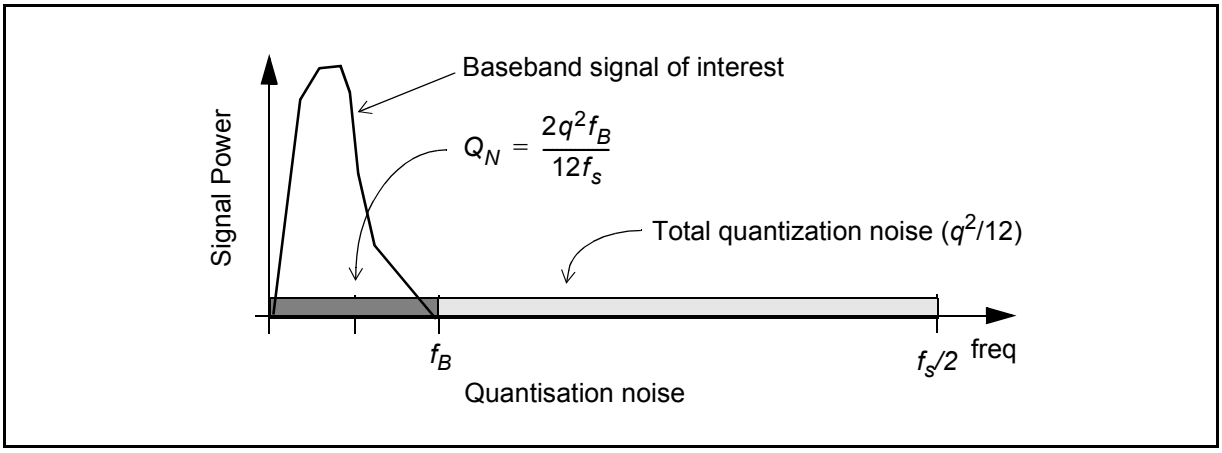
If an *oversampled* digital audio signal is input to a DSP processor, clearly the processing rate must now run at the *oversampled* rate. This requires R x's the computation of its Nyquist rate counterpart (i.e. the impulse response length of all digital filters is now increased by a factor of R), and at a frequency R x's higher. Hence the DSP processor may need to be R x's faster to do the same useful processing as the baseband sampled system. This is clearly not very desirable and a considerable disadvantage compared to the Nyquist rate system. Therefore the *oversampled* signal is decimated to the Nyquist rate, first by digital low pass filtering, then by downsampling. Therefore any frequencies that thereafter exist between 18 and 96 kHz can be removed with a digital low pass filter prior to downsampling by a factor of 4. Hence the complexity of the analogue low pass anti-alias filter has been reduced by effectively adding a digital low pass stage of anti-alias filtering.

For an R x's *oversampled* signal the only portion of interest is the baseband signal extending from 0 to $f_n/2$ Hz, where f_n is the Nyquist rate and $f_s = Rf_n$, and hence the decimation described above is required. Therefore in order to reduce the processing rate to the baseband rate the oversampled signal is first digitally low pass filtered to the $f_n/2$ using a digital filter with a sharp cut-off. The

resulting signal is therefore now bandlimited to $f_n/2$ and can be downsampled by retaining only every R-th sample. This process of oversampling has therefore reduced the specification of the analog anti-alias filter, by introducing what is effectively a digital anti-alias filter. The design trade-off is the cost of the sharp cut-off digital low pass (decimation) filter versus the cost of the sharp cut-off analogue anti-alias filter.

As well as reducing the cost, oversampling can be used to increase the resolution of an ADC or DAC. For example, if an ADC has a quantization level of q volts the in band quantization noise power can be calculated as:

$$Q_N = \frac{2q^2 f_B}{12 f_s} \tag{443}$$



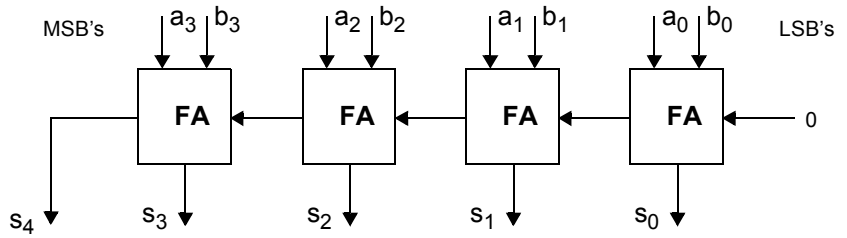
Therefore in order to increase the baseband signal to quantisation noise ratio we can either increase the number of bits in the ADC or increase the sampling rate f_s a number of factors above Nyquist. From the above figure it can be seen that oversampling a signal by a factor of 4 x's the Nyquist rate reduces the in-band quantization noise (assumed to be a flat spectrum between 0 Hz and $f_s/2$ Hz) by 1/4. This noise power is equivalent to an ADC with step size $q/2$ and hence baseband signal resolution has been increased by 1 bit [8]. In theory, therefore, if a single bit ADC were used and oversampled by a factor of 4^{15} ($\approx 10^9 \times f_s$) then a 16 bit resolution signal could be realized! Clearly this sampling rate is not practically realisable. However at a more intuitively useful level, if an 8 bit ADC converter was used to oversample a signal by a factor of 16x's the Nyquist rate, then when using a digital low pass filter to decimate the signal to the Nyquist rate, approximately 10 bits of meaningful resolution could be retained at the digital filter output. See also *Decimation, Noise Shaping, Quantisation Error, Sigma Delta, Upsampling, Undersampling.*

P

P*64: Another name for the H.261 image compression/decompression standard.

Packet: A group of binary digits including data and call control signals that is switched by a telecommunications network as a composite whole.

Parallel Adder: The *parallel adder* is composed of *N* full adders and is capable of adding two *N* bit binary numbers to realise an *N*+1 bit result. A four bit *parallel adder* is:



General 4 bit addition:

$$\begin{array}{rcl} & a_3\ a_2\ a_1\ a_0 & A \\ + & b_3\ b_2\ b_1\ b_0 & B \\ \hline & s_4\ s_3\ s_2\ s_1\ s_0 & S \end{array}$$

Example:

$$\begin{array}{rcl} & 1101 & 13 \\ + & 1011 & 11 \\ \hline & 11000 & \end{array}$$

Four bit binary addition can be performed using a simple linear array of full adder logic circuits. For an *N* bit full adder, *N* full adders are required.

Because the above carry ripples from the LSB to the MSB (right to left) it is often called a ripple adder. The latency of the adder is calculated by finding the longest path through the adder. The above example is for simple unsigned arithmetic, however the *parallel adder* can easily be converted to perform in 2's complement arithmetic [20].

In general inside a DSP processor, the *parallel adder* will be integrated with the parallel multiplier and arithmetic logic unit, thereby allowing single cycle adds, and single cycle multiply-add operations. See also *Arithmetic Logic Unit, Full Adder, Parallel Multiplier, DSP Processor*.

Parallel Multiplier: The key arithmetic element in all DSP processors is the *parallel multiplier* which is essentially a digital logic circuit that allows single clock cycle multiplication of *N* bit binary numbers, where *N* is the wordlength of the processor. Consider the multiplication of two unsigned 4 bits numbers:

General 4 bit multiplication:

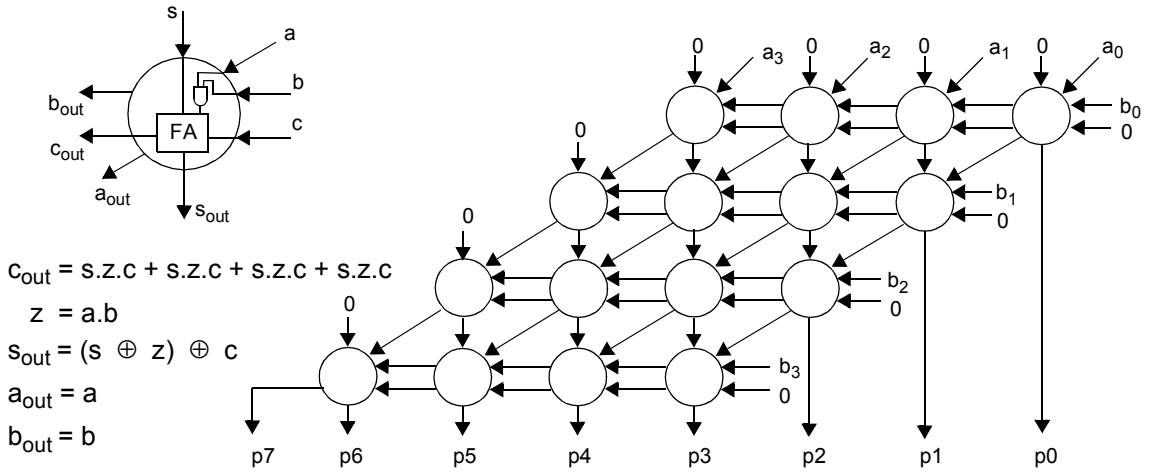
$$\begin{array}{rcl} & a_3a_2a_1a_0 & A \\ \times & b_3b_2b_1b_0 & B \\ \hline & c_3c_2c_1c_0 & \\ & d_3d_2d_1d_0 & \\ & e_3e_2e_1e_0 & \\ & f_3f_2f_1f_0 & \\ \hline & p_7p_6p_5p_4p_3p_2p_1p_0 & P = B \times A \end{array}$$

Example:

$$\begin{array}{rcll} & 1101 & 13 & \\ \times & 1011 & 11 & \\ \hline & 1101 & & \\ & 1101 & & \\ & 0000 & & \\ & 1101 & & \\ \hline & 10001111 & & 143 = 11 \times 13 \end{array}$$

Binary multiplication can be performed using the same partial product formation as used for decimal multiplication. This calculation can then be easily mapped onto an array of full adders with single bit multiplication performed by a simple AND gate.

(In practice 2's complement multiplication is required in DSP calculations to represent both positive and negative numbers, however for the illustrative purpose here the unsigned *parallel multiplier* should suffice; the 2's complement multiplier requires only minor modification [20]). The above 4 bit calculation can be mapped onto an array of binary adders/AND gates:



Each cell of the parallel multiplier has a full binary adder and a logical AND gate. The multiplier performs a binary multiplication by forming the partial products and summing them together using the same mechanism as used in decimal. This multiplier is for positive integer values. Some modification is required to produce a multiplier the operates on 2's complement arithmetic as required for DSP.

The above 4 bit multiplier produces an 8 bit product and requires $4^2 = 16$ cells. Therefore a 16 bit multiplier requires $16^2 = 256$ cells and produces a 32 bit product, and a 24 bit multiplier requires $24^2 = 576$ cells and produces a 48 bit product, and so on. Given that about 12 logic gates may be required for each cell in the multiplier, and each gate requires say 5 transistors, the total transistor count and therefore silicon area required for the multiplier can be very high in terms of percentage of the total DSP processor silicon area. Most general purpose processors do not have *parallel multipliers* and will perform multiplication using the processor ALU and form one partial product per clock cycle, to produce the product in *N* clock cycles (where *N* is the data wordlength).

For some ASIC DSP designs a *parallel multiplier* may be too expensive and therefore a bit serial multiplier may be implemented. These devices require only *N* cells, however the latency is *N* clock cycles [12]. See also *Division, DSP Processor, Full Adder, Parallel Adder, Square Root*.

Parallel Processing: When a number of DSP processors are connected together as part of the same system, this is referred to as *parallel processing* system, as the DSPs are operating in parallel. Although defined as a research area on its own (for complex parallel systems), some simple *parallel processing* approaches to decomposing DSP algorithms are usually rather obvious where small numbers of DSPs are concerned.

Parseval's Theorem: The total energy in a signal can be calcuated based on its time representation, or its frequency representation. Given that the power calculated in both domains must be the same, this equality is called *Parseval's theorem*.

From the Fourier series, recall that a signal, $x(t)$, can be represented in terms of its complex Fourier series:

$$\begin{aligned} x(t) &= \sum_{n=-\infty}^{\infty} C_n e^{jn\omega_0 t} && \text{Synthesis} \\ C_n &= \frac{1}{T}\int_0^T x(t) e^{-jn\omega_o t} dt && \text{Analysis} \end{aligned}$$

Complex Fourier Series Equations (444)

The power in the signal, $x(t)$, can be calculated by integrating over one time period, T:

$$P = \frac{1}{T}\int_0^T |x^2(t)| dt \tag{445}$$

However if we calculated the power based on the power of each of the complex exponential signals, then the total power is:

$$P = \sum_{n=-\infty}^{\infty} \left|C_n e^{jn\omega_0 t}\right|^2 = \sum_{n=-\infty}^{\infty} |C_n|^2 \left|e^{jn\omega_0 t}\right|^2 = \sum_{n=-\infty}^{\infty} |C_n|^2 \tag{446}$$

given that power in the complex exponential $e^{jn\omega_0 t} = \cos n\omega_0 t + j\sin n\omega_0 t$ is 1. Hence for the complex Fourier series representation of a signal, we can state *Parseval's theorem* as:

$$\frac{1}{T}\int_0^T x^2(t) dt = \sum_{n=-\infty}^{\infty} |C_n|^2 \tag{447}$$

If the periodic signal $x(t)$ is real valued, we can also stated *Parseval's theorem* in terms of the amplitude/phase Fourier series representation. Recalling that for a period signal that:

$$\begin{aligned} x(t) &= \sum_{n=0}^{\infty} M_n \cos(n\omega_0 t - \theta_n) \\ \theta_n &= \tan^{-1} B/A \\ M_n &= \sqrt{A_n^2 + B_n^2} \end{aligned} \tag{448}$$

where A_n and B_n are the Fourier coefficients then:

$$P = \sum_{n=0}^{\infty} (M_n \cos(n\omega_0 t - \theta_n))^2 = \sum_{n=0}^{\infty} \frac{M_n^2}{2} \tag{449}$$

and Parseval's theorem can be stated as:

$$\frac{1}{T}\int_0^T x^2(t)dt = \sum_{n=0}^{\infty} \frac{M_n^2}{2} \tag{450}$$

If a signal is aperiodic, the Parseval's theorem can be stated in terms of the total energy in the signal being the same in the time domain and frequency domain:

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(f)|^2 df \tag{451}$$

See also Discrete Fourier Transform, Fourier Series, Fourier Transform.

Passband: The range of frequencies that pass through a filter with very little attenuation. See also *Filters*.

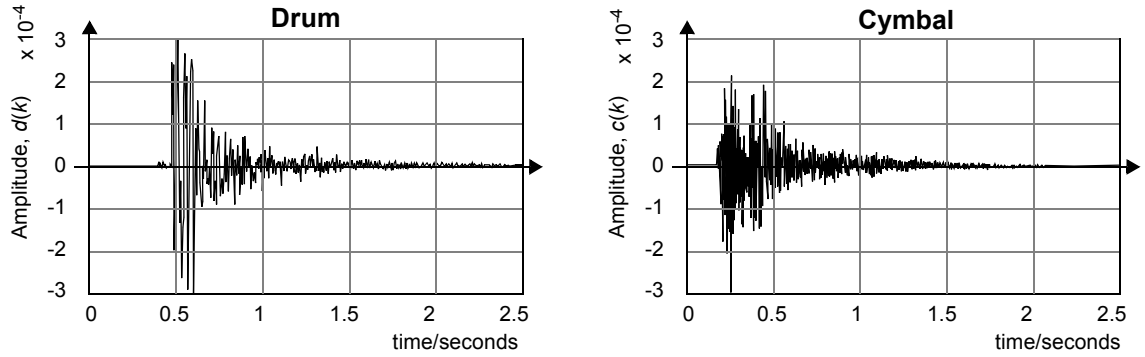
PC-Bus: Plug in DSP cards (or boards) for IBM PC (AT) and compatibles conform to the PC-Bus standard. Through the PC-Bus, a DSP processor will be provided with power, (12V and 5V), Ground lines, and a 16 bit data bus for transfer between DSP board and PC. See also *DSP Board*.

Percentage Error: See *Relative Error*.

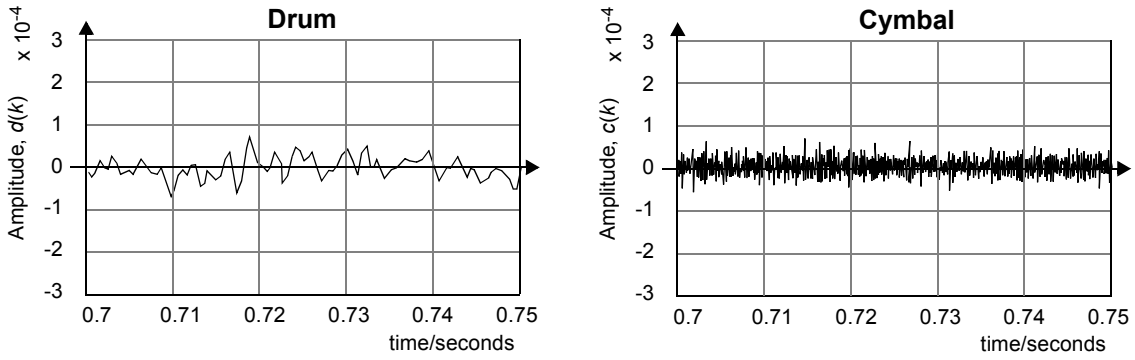
Perceptual Audio Coding: By exploiting well understood psychoacoustic aspects of human hearing, data compression can be applied to audio thus reducing transmission bandwidth or storage requirements [30], [52]. When the ear is perceiving sound, spectral masking or temporal masking may occur - a simple example of spectral masking is having a conversation next to a busy freeway where speech intelligibility will be reduced as certain portions of the speech are masked by noisy passing vehicles. If a perceptual model can be set up which has similar masking attributes to the human ear, then this model can be used to perform perceptual audio coding, whereby redundant sounds (which will not be perceived) do not require to be coded or can be coded with reduced precision. See also *Adaptive Transform Acoustic Coding, Audiology, Auditory Filters, Precision Adaptive Subband Coding (PASC), Psychoacoustics, Spectral Masking, Temporal Masking, Threshold of Hearing.*

Percussion: Any instrument which can be struck to produce a sound can be described as percussive [14]. *Percussion* sounds are either pitched or unpitched. For example drums and cymbals are usually unpitched instruments used to create and sustain the rhythm of music. Certain type of drums however, such as timpani actually have an associated pitch. Xylophones and marimba's are pitched *percussion* instruments with a range of three or four octaves.

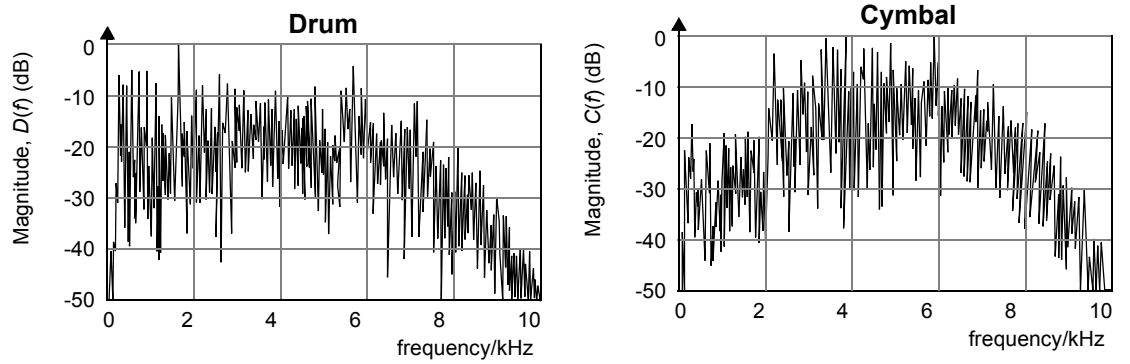
In the figures below the sound pressure level volume envelope, a short time segment and a frequency domain representation is shown for a cymbal strike and a snare drum beat.



The variation in sound pressure level for a drum beat and cymbal strike. Both signals last for about 1.5 seconds. From a simple visual inspection the cymbal seems to have more sustain and is a “fuller” waveform.



A short 0.15 second segment of the drum and cymbal signals clearly shows the cymbal to contain a wider range of higher frequencies. Both signals are random in nature with little discernible periodic content.



Taking an FFT over a short 0.05 segment of the drum and cymbal waveforms serves to illustrate the stochastic nature of the two sounds.

From the above figures it can be seen that the drum beat and cymbal strike signals both appear to be stochastic in nature although given that they produce sound based on a resonating impulse there is clear quasi-periodic content. These signals also possess a degree of regularity in that successive

strikes sound “similar”. The drum exhibits a lower frequency content than the cymbal which is consistent with the more “bassy” sound it has.

The sound pressure level created by drums and cymbals depends on the force with which they are struck; both are capable of generating up to 100 dB at a distance of 1 metre. See also *Music, Western Music Scale*.

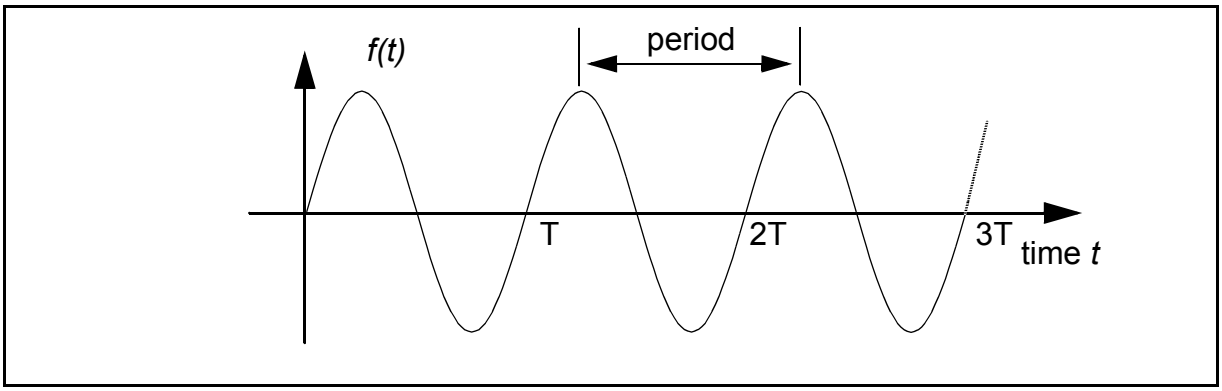
Perfect Pitch: The ability to exactly specify the name of a musical note being played on the Western music scale is called *perfect pitch*. Only a very few individuals have *perfect pitch*, and there is still some debate to whether such skills can be learned. Many individuals and musicians have good relative pitch, whereby given the name of one note in a sequence, they can correctly identify others in the sequence. See also *Music, Pitch, Relative Pitch, Western Music Scale*.

Permanent Threshold Shift (PTS): When the threshold of hearing is raised due to exposure to an excessive noise a *permanent threshold shift* is said to have occurred. See also *Audiology, Audiometry, Temporary Threshold Shift (TTS), Threshold of Hearing.*

Permutation Matrix: See *Matrix Structured - Permutation.*

Period: The period, *T*, of a simple sine waveform is the time it takes for one complete wavelength to be produced. The inverse of period, gives the frequency, or the number of wavelengths in one sec:

$$f = \frac{1}{T} \tag{452}$$



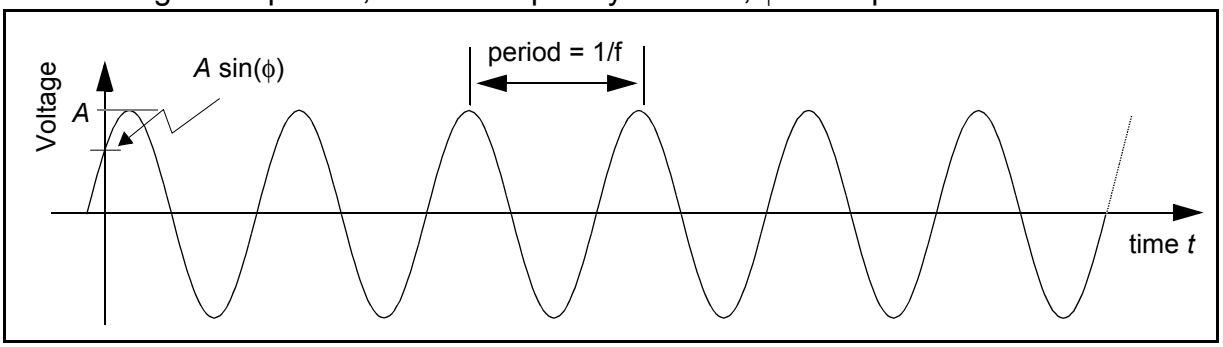
Personal Computer Memory Card International Association (PCMCIA): The name given to bus slots that became almost standard on notebook and subnotebook PCs around 1994. PCMCIA cards were originally memory cards, but now modems, small disk drives, digital audio soundcards, and DSP cards are available. The term PC Card is now being used in preference to the rather unwieldy acronym PCMCIA [169].

Personal Digital Assistant (PDA): A consumer electronics category which classifies handheld computers that can decode handwritten information (pattern recognition) and communicate with other computers and FAX machines [169].

Phase: The relative starting point of a periodic signal, measured in angular units such as radians or degrees. Also, the angle a complex number makes relative to the real axis. A *sine* wave (occurring with respect to time) can be written as:

$$x(t) = A\sin(2\pi ft + \phi) \tag{453}$$

where *A* is the signal amplitude; *f* is the frequency in Hertz; ϕ is the phase and *t* is time.



Phase Compensation: A technique to modify the phase of a signal, but leaving the magnitude response unchanged. *Phase compensation* is usually peformed using an all-pass filter. If the phase of a system is compensated to produce an overall linear phase, then this is often refered to as group delay equalisation as linear phase corresponds to a constant group delay. See *All-pass Filter-Phase Compensation, Equalisation, Finite Impulse Reponse Filter - Linear Phase*.

Phase Delay: A term usually synonymous with group delay. See *Group Delay*.

Phase Jitter: In telephony the measurement (in degrees out of phase) that an analog signal deviates from the referenced phase of the main data carrying signal. Phase jitter interferes with the interpretation of information by changing the timing or misplacing a demodulated signal in frequency. See also *Clock Jitter*.

Phase Modulation: One of the three ways of modulating a sine wave signal to carry information. The sine wave or carrier has its phase changed in accordance with the information signal to be transmitted. See also *Amplitude Modulation, Frequency Modulation*.

Phase Response: See also *Fourier Series - Amplitude/Phase Representation, Fourier Series - Complex Exponential Representation*.

Phase Shift Keying (PSK): A digital modulation technique in which the information data bits are encoded in the phase of the carrier signal. The receiver recovers the data bits by detecting the phase of the received signal over a symbol period and decoding this phase into the appropriate data bit pattern. See also *Amplitude Shift Keying, Differential Phase Shift, Frequency Shift Keying.*

Phasing: A musical effect whereby the phase of a signal is modified, mixed (or added) with original signal, and the composite signal is then played [32]. See also *Music, Music Synthesis*.

Phons: The *phon* (pronounced *fone*) is a (subjective) measure of loudness. The units of *phons* are given to the sound pressure level of a 1000Hz tone that a human listener has judged to be equally loud to the sound to be measured. Hence to measure a particular sound in *phons* would require a listener to switch back and forth between a calibrated, variable 1000Hz tone and the sound to be measured. See also *Equal Loudness Contours, Equivalent Sound Continuous Level, Frequency Range of Hearing, Sound Pressure Level*.

Piezoelectric: Piezoelectric materials can convert mechanical stress into electrical output energy, hence they are widely used as sensors. Piezoelectric crystals are also used in a feedback configuration to make very precise clocks.

Pipelining Execution: DSP processors having RISC architectures often implement a pipelining structure whereby instructions are executed by the processor in four stages: (1) Instruction Fetch,

(2) Instruction Decode, (3) Memory Read, (4) Execute. Each stage takes one cycle of the processor clock, meaning that each instruction is a minimum of 4 clock cycles. However because the DSP processor has been designed to be *pipelined*, the processor can perform all four stages in one cycle. Hence this overlapping means that on average one instruction can be executed every clock cycle.

Pink Noise: *Pink noise* is similar to white noise, except that rather than having a flat power spectrum, it falls off at 10dB/decade. Pink noise is sometimes referred to a $1/f$ noise.

Pitch: There are a number of varying definitions of pitch, however the generic meaning is the subjective quality of a sound which positions it somewhere in the musical scale [14]. As the number of cycles per second of a musical note increases linearly our perceived sense of pitch increases logarithmically. Although very similar to frequency which is measured exactly, pitch is determined subjectively. For example if two pure tones of slightly different frequencies are presented to a listener and they are allowed to adjust the intensity levels of one of them, then it is likely that they will be able to find a level where both tones *sound* as if they have the same pitch. Pitch is therefore to some extent dependent on intensity. At louder levels for low frequency tones the pitch decreases with increase in intensity, but for high tones the pitch increases with increase in intensity. See also *Music, Perfect Pitch, Western Music Scale*.

Pivotting: See *Matrix Decompositions - Pivoting*.

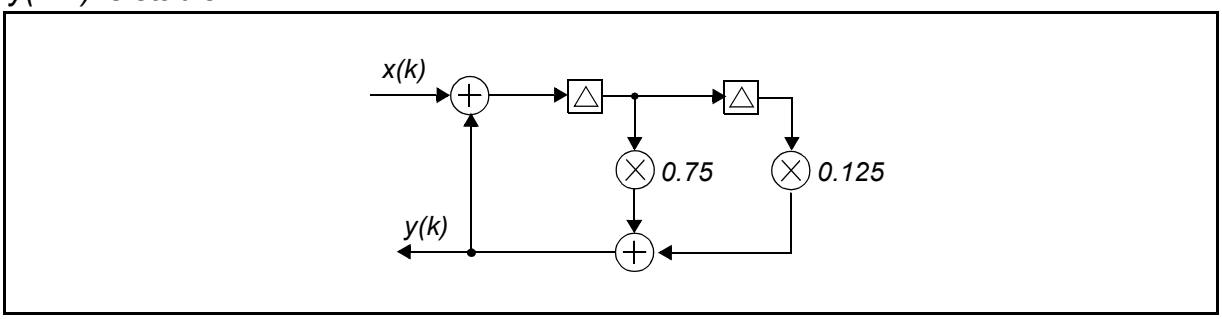
Plane Rotations: See *Matrix Decompositions - Plane Rotations*.

Plosives: One of the elementary sounds of speech, namely plosives, fricatives, sibilant fricative, semi-vowels, and nasals. Plosives are formed by blocking the vocal tract so that no air flows and suddenly removing the obstruction to produce a puff of air. Examples of plosive sounds are “p”, “b”, “t”, “d”, “g”, and “k”. See also *Fricatives, Nasals, Semi-vowels,* and *Sibilant Fricatives*.

PN Sequence: See *Pseudo-Random Noise Sequence.*

Polar: *Polar* refers to the type of signalling method used for digital data transmission, in which the marks (ones) are indicated by positive polarities and the spaces (zeros) are indicated by negative polarities (or vice-versa). See also *Bipolar (2), Non-return to Zero.*

Poles: If the impulse response of a recursive system (with feedback) is transformed into the z-domain, the *poles* of the function are found by factoring the denominator polynomial to find the roots. If the poles are outside the unit circle, then this is an indication that the system is unstable. The transfer function *H(z)* of a simple two pole IIR filter with the output *y(n) = x(n) + 0.75y(n-1) - 0.125 y(n-2)* is stable:



$$H(z) = \frac{1}{(1 - 0.75z^{-1} + 0.125z^{-2})} = \frac{1}{(1 - 0.5z^{-1})(1 - 0.25z^{-1})} \quad (454)$$

i.e. the poles are $z = 0.25$ and $z = 0.5$. If the roots were outside of the unit circle (having a magnitude greater than 1), then the system, *h(n)* would be unstable.

Positive Definite Matrix: See *Matrix Properties - Positive Definite.*

Positive Semi-definite: See *Matrix Properties - Positive Semi-definite.*

Postmultiplication: See *Matrix Operations - Postmultiplication*

Power Spectral Density (PSD): The *power spectral density* describes the frequency content of a stationary stochastic or random signal. The *PSD* can be estimated by taking the average of the magnitude squared DFT sample values (the periodogram). Many other DSP techniques have been developed for estimating signal frequency content. This area of research is collectively call spectral estimation. The *PSD* is calculated from the Fourier transform of the autocorrelation function:

$$\text{Power Spectral Density, } S(f) = \sum_{n=-\infty}^{\infty} r(n)e^{-j2\pi fn} \quad (455)$$

where the autocorrelation function, $r(n)$, provides a measure of the predictability of a signal, $x(k)$:

$$r(n) = E\{x(k)x(k+n)\} = \sum_{k} x(k)x(k+n)p\{x(k), x(k+n)\} \quad (456)$$

where $p\{x(k), x(k+n)\}$ is the joint probability density function of $x(k)$ and $x(k+n)$. For signals assumed to be ergodic the *autocorrelation* can be estimated as a time average:

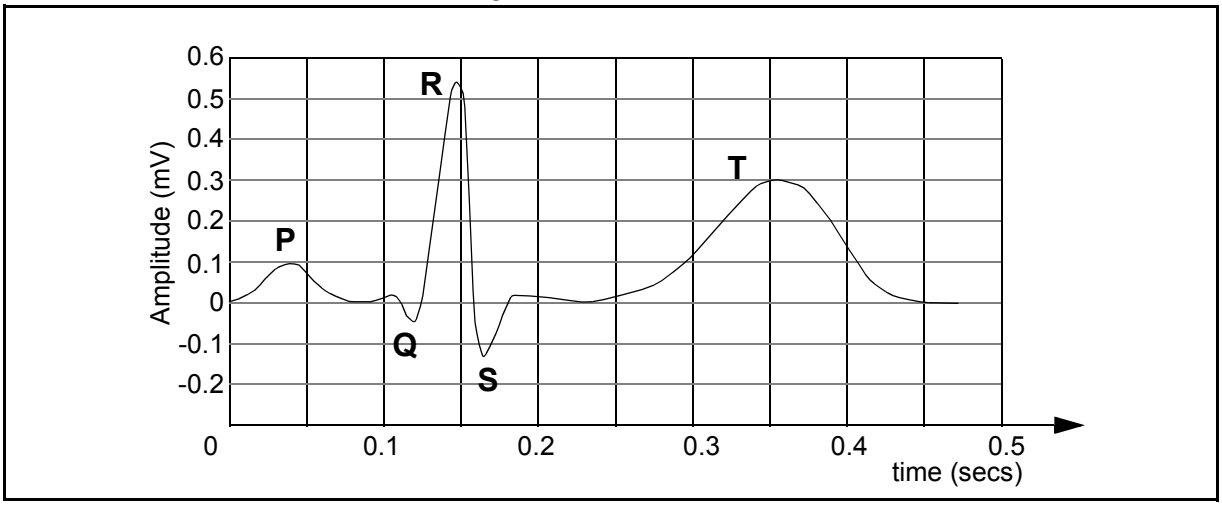
$$r(k) = \frac{1}{2M-1} \sum_{k=0}^{2M-1} x(n)x(n+k) \quad \text{for large } M \quad (457)$$

If a particular autocorrelation function is estimated for *n* different time lags, then a *PSD* estimate can be computed as the DFT of these correlations.

.See also *Autocorrelation, Discrete Fourier Transform*.

Power Rails: The voltage used to power a DSP board will usually consist of a number of voltage sources, which are often referred to as power rails. For a DSP board, there are usually digital power rails (0 volts and 5 volts) to power the digital circuitry, and analog power rails (-12 volts, 0 volts, and +12 volts) to power the analog circuitry.

PQRST Wave: The name given to the characteristic shape of an electrocardiogram (heartbeat) signal waveform. See also *Electrocardiogram.*

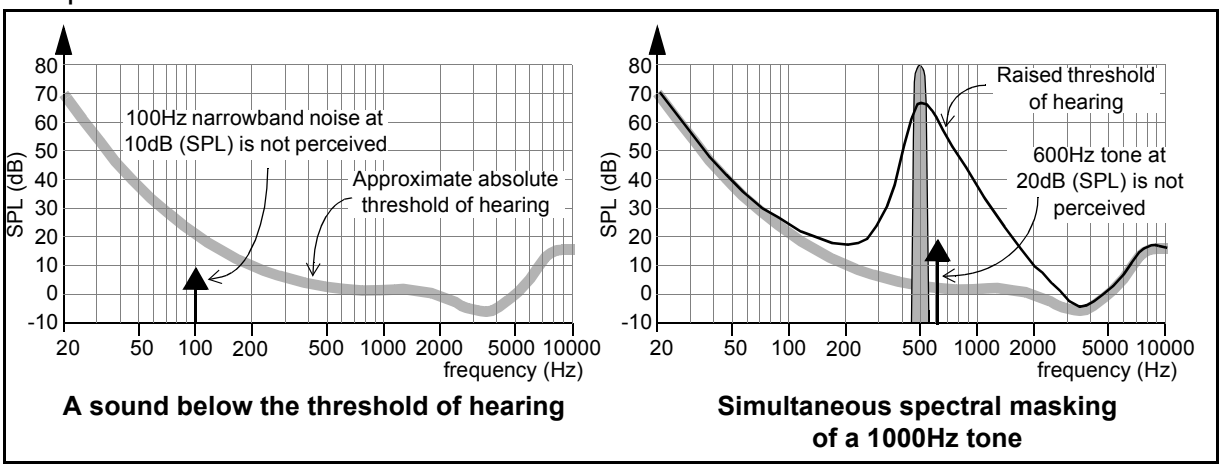


Precedence Effect: In a reverberant environment the sound energy received by the direct path can be much lower than the energy received by indirect reflected paths. However the human ear is still able to localize the sound location correctly by localizing the first components of the signal to arrive. Later echoes arriving at the ear increase the perceived loudness of the sound as they will have the same general spectrum. This psychoacoustic effect is known as the *precedence effect*, law of the first wavefront, or sometimes the Haas effect. The *precedence effect* applies mainly to short duration sounds or those of a discontinuous or varying form. See also *Ear, Lateralization, Source Localization, Threshold of Hearing.*

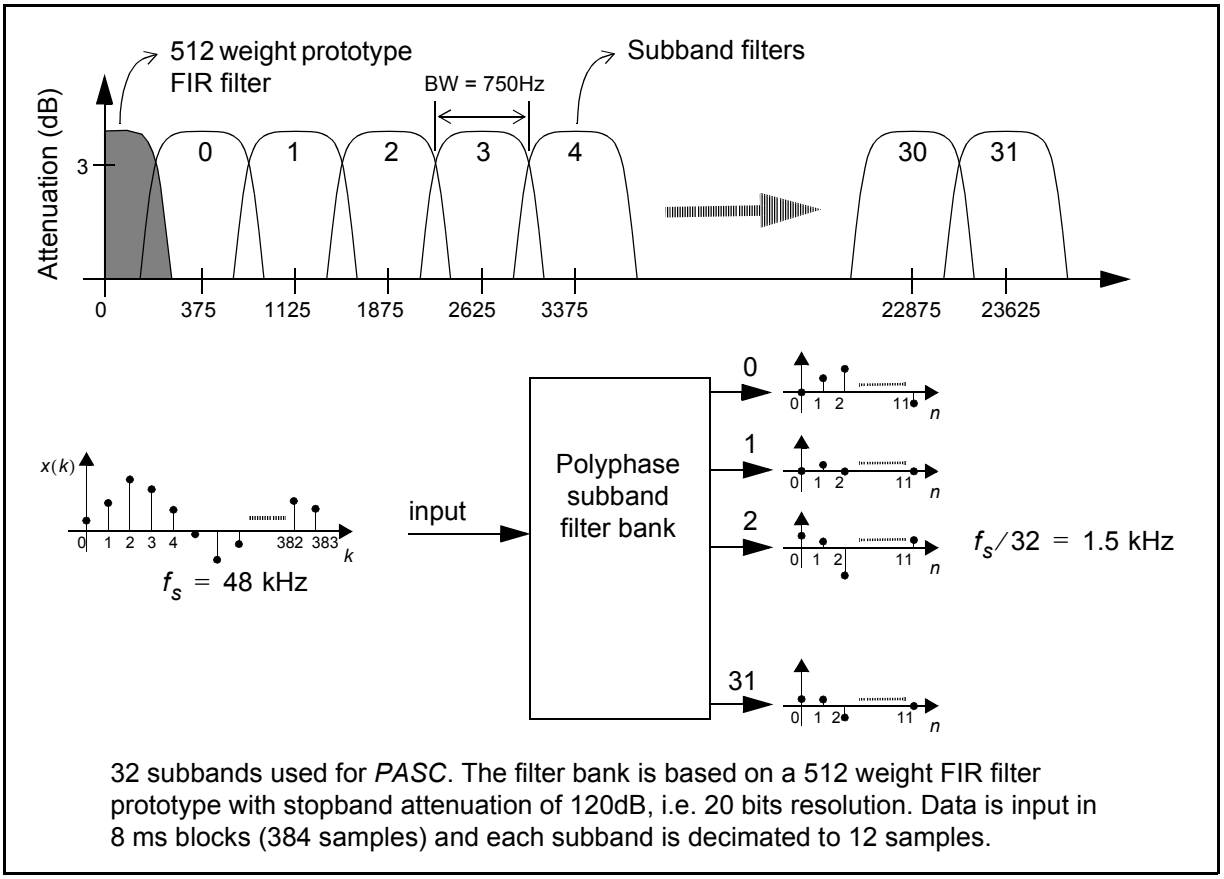
Precision Adaptive Subband Coding (PASC): A data compression technique developed by Philips and used in hifidelity digital audio systems such as digital compact cassette (DCC). PASC is closely related to the audio compression methods defined in ISO/MPEG layer 1. Listening tests have revealed that the overall quality of *PASC* encoded music is "*almost identical to that of compact disc (CD)*". In fact it has been argued that in terms of dynamic range DCC has improved performance given that it is compressing 20 bit PCM data compared to the encoding of 16 bit PCM data by a CD [83].

Precision adaptive subband coding compresses audio by **not** coding elements of an audio signal that a listener will not hear. PASC is based mainly on two psychoacoustic principles. First, the ear only hears sounds above the absolute threshold of hearing, and therefore any sounds below this threshold do not require to be coded. Second louder sounds spectrally mask quieter sounds of a "similar" frequency such that the quiet sound is unheard in the simultaneous presence of the louder

sound due to the psychoacoustic raising of the threshold of hearing. The following figure illustrates both principles:



In order to exploit psychoacoustic masking the first stage of a PASC system splits the Nyquist bandwidth of a signal (of between 16 and 20 bit resolution) sampled at 48kHz into 32 equal subbands each of bandwidth 750Hz. This is accomplished using a 512 weight prototype FIR low pass filter, $h(n)$, of 3dB bandwidth 375Hz, and stopband attenuation 120dB. Note that to achieve 120dB attenuation 20 bit filter coefficients are required. By modulating the impulse response $h(n)$ with modulating frequencies of 375Hz, 1125Hz, 1875Hz and so on in 750Hz intervals, a series of 32 bandpass filters with a 3dB bandwidth of 750Hz and centered around the modulating frequency are produced. A polyphase subband filter bank is therefore set up as illustrated below



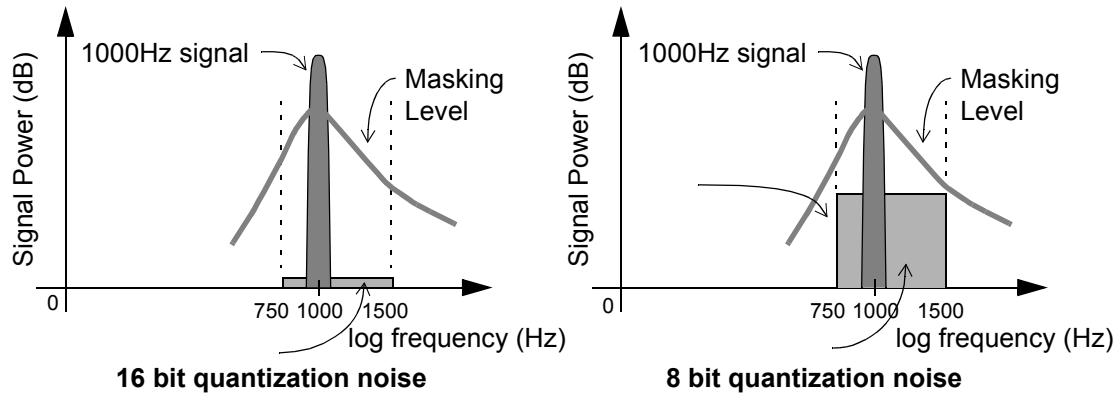
32 subbands used for *PASC*. The filter bank is based on a 512 weight FIR filter prototype with stopband attenuation of 120dB, i.e. 20 bits resolution. Data is input in 8 ms blocks (384 samples) and each subband is decimated to 12 samples.

(Note that although aliasing occurs between adjacent subbands, the alias components are cancelled when the subbands are merged to reconstruct the original audio data spectrum [49].) The

input data stream is subband filtered in blocks of 8 ms, which corresponds to 384 samples (48000×0.008). Therefore the output of each subband filter after decimation consists of 12 samples.

With the signal in subband coded form the second stage of the *PASC* system is to perform a comparison of the full audio spectrum with a model of the human ear. The subband filtering allows a simple (but coarse) spectral analysis of the signal to be produced by calculating the power of the 12 sample values in each subband. If the power in a subband is below the threshold of hearing, then the subband is treated as being empty and does not need to be coded for the particular 8ms block being analyzed. If the power in a particular subband is above the threshold of hearing then a comparison is made with the known masking threshold to calculate the in-band masking level. Following this the level of masking caused by this signal in other neighboring subbands is established. The overall masking calculation is accomplished using a 32×32 matrix containing the masking information and defined in the ISO/MPEG standard.

From the masking calculation results, a decision is made as to the number of bits that will be allocated to represent the data in that subband such that the quantization noise introduced is below the masking level (or raised threshold of hearing) and will therefore not be heard when the audio signal is reconstructed. The bit rate of a *PASC* encoded time frame of 8ms is fixed at 96 bits/frame (for each subband, on average). Therefore the bits must be allocated judiciously to the subbands. The subbands with the highest power relative to the masking level are allocated first as it is likely they will be important and dominant sounds in the overall audio spectrum and will require the best resolution. If two subbands have the same ratio, the lower frequency subband is given priority over the higher one. An example of quantization noise masking is given below:



The 1000 Hz narrowband noise will spectrally mask any signals below the masking level (or raised threshold of hearing). Therefore, considering only this subband, when the signal is reproduced the higher level of quantization noise in the 8 bit signal will not be perceived. Hence the 8 bit signal has the same perceived quality as the 16 bit signal and data compression has been achieved without noticeable loss in quality. Note the masking effect of signals in nearby subbands may extend into the 750-1500Hz subband which could further increase the masking level and therefore allow even fewer bits to represent the signal.

Rather than fixed point sample values (as used in the above illustrative example) *PASC* uses a simple block floating point number representation to represent sample values. The mantissa can be between 2 and 15 bits and the exponent is a 6 bit value. The actual number of bits assigned to the mantissa depend on the masking calculations. This leads to an overall dynamic range from

+6dB to -118dB (the extra 6dB headroom is required due to the subband filtering process) which is more than the 96dB available from 16 bit linear coding.

On average a *psychoacoustic subband coded* music signal rarely assigns bits to subbands covering the frequency range 15 kHz to 24 kHz (i.e. they are usually empty!), around 3 to 7 mantissa bits will typically be required for subbands covering the frequency range 5kHz - 15 kHz, and for the frequency range 100Hz - 5 kHz between 8 and 15 bits are typically required. The higher bit allocation for lower frequencies is as expected as the masking effect is less pronounced at lower frequencies (see *Spectral Masking*). This allocation of precision would perhaps suggest that the initial subband structure should have a small bandwidth for low frequencies and a higher bandwidth for larger frequencies. However the small bandwidth required at low frequencies would require a very long impulse response filter which needs to be compensated for by delaying the output signal from higher subbands which have a smaller bandwidth if phase is to be preserved. To implement this delay on chip requires such a large area that this solution is not economically attractive, albeit good compression ratios would be possible.

After each 8 ms time frame has undergone the *PASC* coding and bit allocation, the data is then stored in a encoded bit stream for recording to magnetic tape. Cross interleaved Reed-Solomon code (CIRC) is used for error correction coding of *PASC* data when recorded onto DCC (digital compact cassette).

PASC techniques can also be applied to input data sampled at 32kHz or 44.1kHz. Because the data rate stays the same at 384bits/sec, the subband filter bandwidth for these sampling frequencies reduces to 500Hz and 698Hz respectively.

See also *Adaptive Transform Acoustic Coding (ATRAC), Auditory Filters, Compact Disc, Data Compression, Digital Compact Cassette (DCC), Frequency Range of Hearing, Psychoacoustics, Spectral Masking, Subband Filtering, Temporal Masking, Threshold of Hearing*.

Premultiplication: See *Matrix Operations - Premultiplication*

Probability: The use of probabilistic measures and statistical mathematics in digital signal processing is very important. Specifically the concept of a random variable which is characterised via a *probability* density function (PDF) is very important. With *probability*, random signals can be characterised and information on their frequency content can be realised.

In its simplest form the *probability* of an event A happening, and denoted as $p(A)$ can be determined by performing a large number of trials, and counting the number of times that event A occurs. Therefore:

$$P(A) = \lim_{n \to \infty} \frac{\text{no. of times A occured, } n_A}{\text{total no. of trials, } n} \tag{458}$$

determines the *probability* of event A occurring. A simple example is the shaking of a die to determine the probability of a 6 occurring. If, for example 60 trials were done and a 6 occurred 8 times then $P_d(6) = 8/60$, where the subscript “d” specifies the process name. Of course the true *probability* is $P_d(6) = 1/6$ which would have been determined if an “infinite” number of trials were done.

From the above simple definition, it can be noted that $0 \le P(A) \le 1$. Clearly if $P(A) = 0$ (the null event) then the event (almost) never occurs, whereas if $P(A) = 1$ then it (almost) always occurs

(the sure event). If you find the paranthetical "almosts" annoying, amusing, confusing, etc., remember that probability means never having to say you're certain (or was that statistics?).

The joint probability that the event *AB* will occur is denoted $P(AB)$. The following definitions are also useful for *probability*:

- **Bayes Theorem:** The joint *probability* that an event *AB* occurs can be expressed as:

$$P(AB) = P(A)P(B|A) = P(B)P(A|B) \tag{459}$$

  If two events *A* and *B* are independent then $P(A|B) = P(A)$ or $P(B|A) = P(B)$.

- **Conditional Probability:** The probability that an event *A* occurs, where an event *B* has already occurred is denoted as $P(A|B)$.

- **Independence:** Two separate events, *A* and *B*, are independent if the probability of *A* and *B* occurring is obtained from the multiplication of the probability of *A* occurring, and *B* occurring:

$$P(AB) = P(A)P(B) \tag{460}$$

- **Joint Probability:** The probability of two events, *A* and *B*, occurring is:

$$P(AB) = \lim_{n \to \infty} \frac{\text{no. of times AB occured, } n_{AB}}{\text{total no. of trials, } n} \tag{461}$$

  where the notation *P(AB)* can be read "the probability of event *A* and event *B*. As an example consider an experiment where a coin is flipped, and a die is shaken at the same time. The probability that a head shows up $P_c(\text{head})$, and the number 3, $P_d(3)$ is:

$$P(\text{head \& } 3) = P_d(3)P_c(\text{head}) = \frac{1}{6} \times \frac{1}{2} = \frac{1}{12} \tag{462}$$

  The shaking of the die and flipping of the coin are both independent events, i.e. the outcome of the coin flip has no bearing on the outcome of the die shake.

See also *Ergodic, Expected Value, Mean Value, Mean Squared Value, Probability, Random Variable, Variance, Wide Sense Stationarity.*

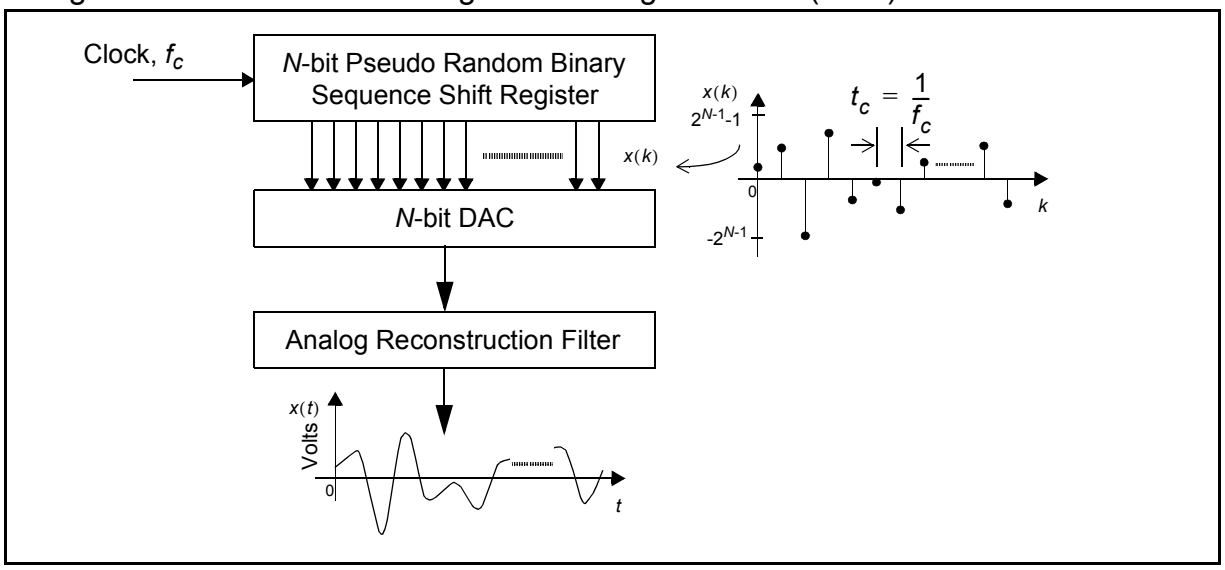
Probability Density Function: See *Random Variable*.

Proportional Integral Derivative (PID) Controller: Process control applications monitor a variable such as temperature, level, flow and so on, and output a signal to adjust that variable to equal some desired value. In a *PID* the difference between the desired and measured variable is found (the error), and if large then the integral part of the controller causes the output to change faster and the derivative adjusts the magnitude of the output (controlling) signal in proportion to the error rate. PID controllers usually do not require the processing power of a DSP as the data processing rates are well within that of microcontrollers.

Pseudo-Inverse: See *Matrix Properties - Pseudo-Inverse*.

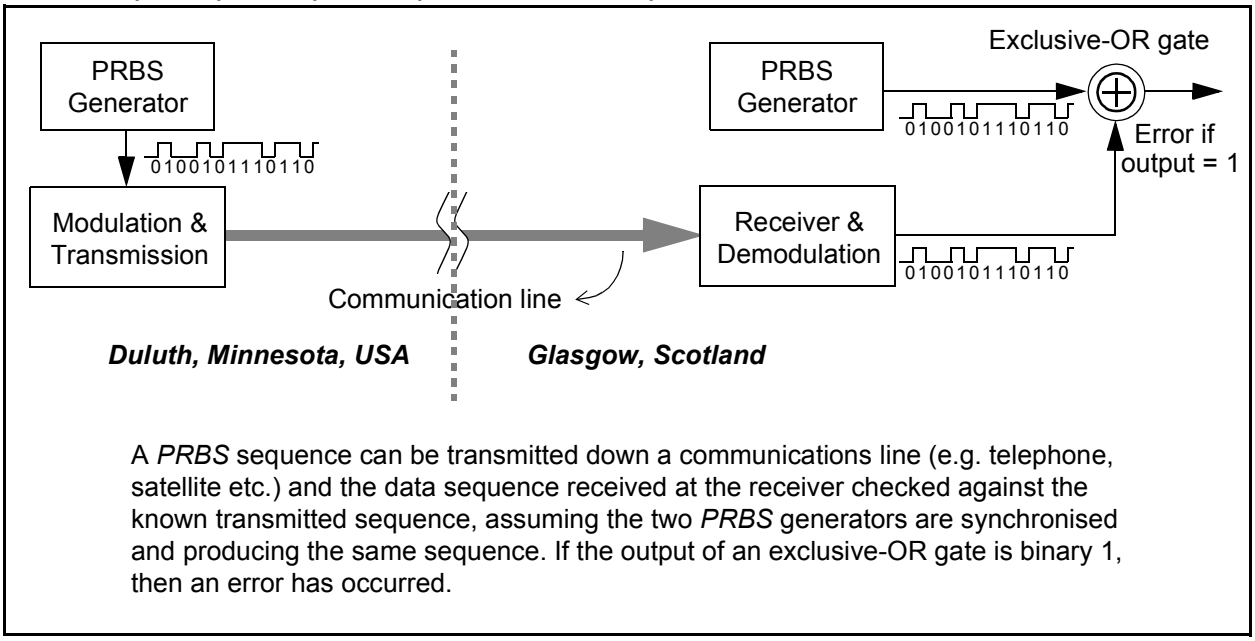
Pseudo-Inverse Matrix: See *Matrix Properties - Pseudo-Inverse.*

Pseudo-Noise (PN): Analog pseudo-noise can be generated using pseudo random binary sequence generator connected to a digital to analog converter (DAC):



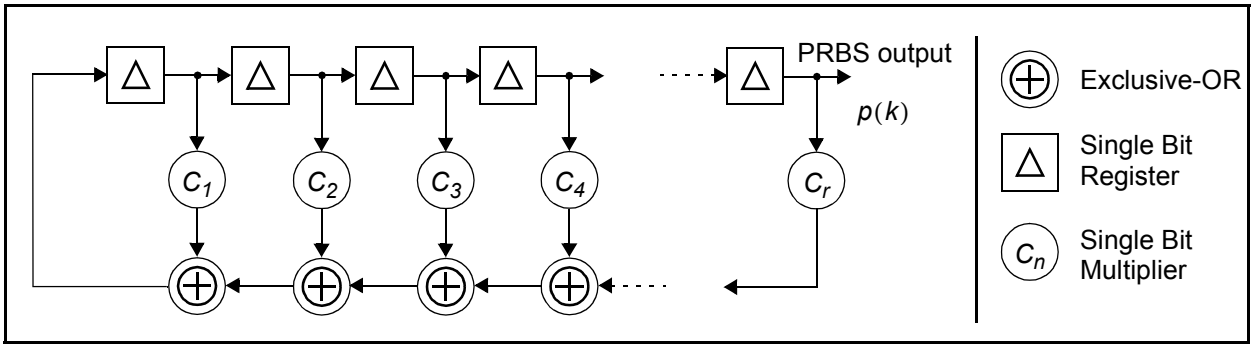
The period of the pseudo noise is Nt_c seconds. There are of course other methods of producing analog "noise", however the term pseudo noise usually indicates that the sequence was generated using pseudo random noise sequence generating schemes. See also *Pseudo-Random Noise Sequence, Pseudo-Random Binary Sequence*.

Pseudo-Random Binary Sequence (PRBS): The *PRBS* is a binary sequence generated by the use of an *r*-bit sequential linear feedback shift register arrangement. *PRBS*'s are sometimes called pseudo noise (PN) sequences and pseudo random noise (PRN). *PRBS*'s are widely used in digital communications, where for example both ends of digital channel contain a circuit capable of generating the same *PRBS*, and which can therefore allow the bit error rate of the channel to be measured, or perhaps adaptive equalization to be performed.


PRBS Generator
0100101110110
Modulation & Transmission
Communication line
Receiver & Demodulation
PRBS Generator
Exclusive-OR gate
0100101110110
Error if output = 1
0100101110110
Duluth, Minnesota, USA
Glasgow, Scotland
A PRBS sequence can be transmitted down a communications line (e.g. telephone, satellite etc.) and the data sequence received at the receiver checked against the known transmitted sequence, assuming the two PRBS generators are synchronised and producing the same sequence. If the output of an exclusive-OR gate is binary 1, then an error has occurred.

Other applications include using *PRBS* for spread spectrum communications [9], for scrambling data, and using a *PRBS* for range finding via radar or sonar [116].

A *PRBS* is called pseudo random because in actual fact the sequence repeats over a large number of bits and is therefore actually periodic, however the short term behaviour of the sequence appears random. The general construction of a *PRBS* producing linear feedback shift register of length *r* bits is:



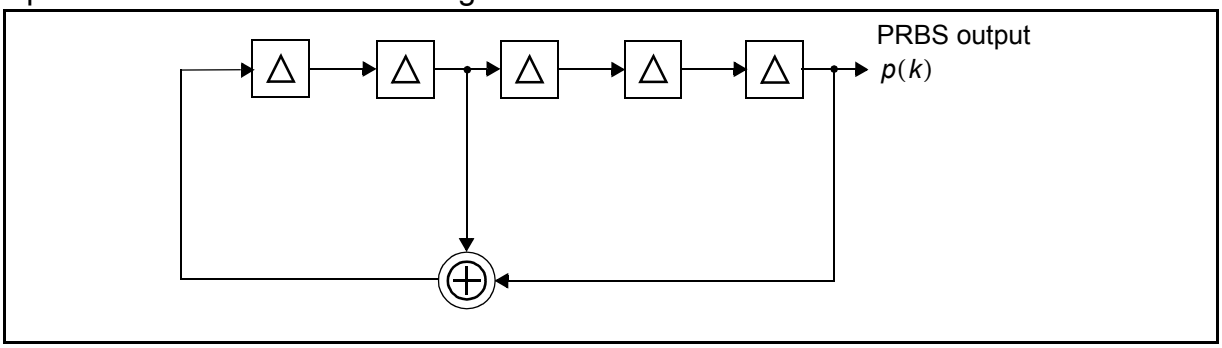
where the register is clocked at every T_c seconds (often denoted as the chip interval), and the binary data signal, $p(k)$, is therefore output at a rate of $f_c = 1/T_c$. The longer the register, then the longer the *PRBS* that can be generated. The values of the single bit multipliers C_r are either 0 or 1 and they can be represented in a convenient characteristic polynomial notation:

$$f(X) = 1 + \sum_{k=1}^{r} C_k X^k = C_r X^r + C_{r-1} X^{r-1} + \ldots + C_1 X + 1 \qquad (463)$$

By carefully choosing the polynomial it is possible to ensure that the shift register cycles through all the possible states (or *N*-tuples), with the exception of the all zero state [40]. This will produce a PRBS of $2^r - 1$ bits (and known as a maximal sequence) before the cycle restarts. If the register ever enters the zero state it will never leave. As an example consider a 31 bit maximal length sequence can be produced from the polynomial:

$$X^5 + X^2 + 1 \qquad (464)$$

which specifies the 5 bit *PRBS* shift register:

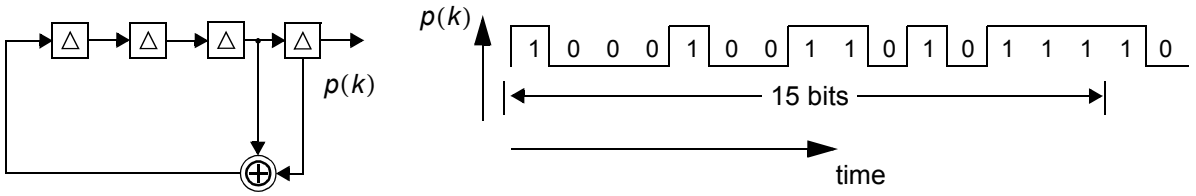


For a particular *PRBS*, a sequence of the same bits (either 1's or 0's) is referred to as "run", and the number of bits in the run, is the "length". For a maximal length sequence from an *r* bit register of length *N* (= $2^r - 1$) bits it can be shown that the *PRBS* will contain one run length of *N* 1's, and one

run of $N-1$ 0's. The number of other run lengths of 1's and 0's increases with the power of 2 as follows:

Run Length	1's	0's
N	1	0
N-1	0	1
N-2	1	1
N-3	2	2
: 3	: 2^{N-5}	: 2^{N-5}
2	2^{N-4}	2^{N-4}
1	2^{N-3}	2^{N-3}

For example an $r = 4$ bit shift register can be set up from the polynomial $X^4 + X^3 + 1$ to produce a 15 bit maximal length *PRBS* as follows:



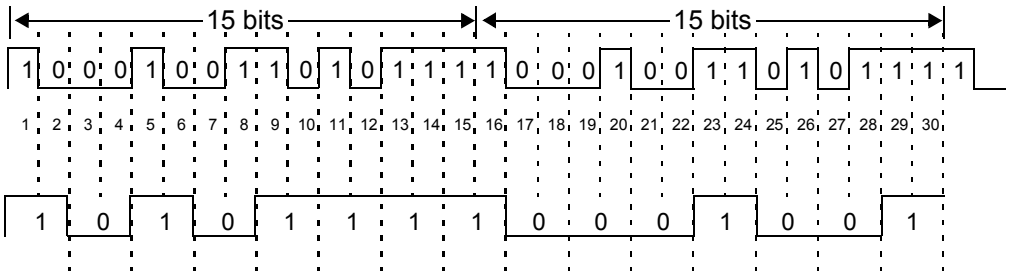
Priming the shift register with 0001, will cause it to cycle through 1000, 0100, 0010, 1001, 1100, 0110, 1011, 0101, 1010, 1101, 1110, 1111, 0111, 0011, and back to 0001. If the contents of the shift register are considered as a binary number, then a *PRBS* generator contains all binary numbers from 1 to 2^{N-1} in a "random" order. Note that the *PRBS* has a sequence of four 1's, one sequence of three 0,s and so on accordance with the above table denoting the run lengths for an *N* bit *PRBS*.

Note that when a *PRBS* is generated over *N* clock cycles, then the shift register contains at some point, all binary numbers from 1 to 2^{r-1}, i.e. except zero, a state from which the *PRBS* can never leave. Feedback taps for some maximal length sequences using longer shift lengths are shown in the table below:

Shift Register Length, *r*	Maximal Code Length, *N*	Maximal Sequence Generating Polynomials
5	31	X^5+X^3+1
8	255	$X^8+X^6+X^5+X^4+1$
10	1023	$X^{10}+X^7+1$
16	65535	$X^{16}+X^{15}+X^{13}+X^4+1$
20	1048575	$X^{20}+X^{17}+1$
24	16777215	$X^{24}+X^{23}+X^{22}+X^{17}+1$

Note that other polynomials can be used to generate other maximal length sequences of *N* bits. The actual number of maximal length generating polynomials can be calculated using prime factor analysis [116].

A useful property of a maximal length sequence is that the alternate bits in a sequence form the same sequence at half of the rate. Consider two runs of the above 15 bit *PRBS* sequence generated from the polynomial $X^4 + X^3 + 1$ and creating a new sequence by retaining only every second bit:



Taking only every alternate bit then the same *PRBS* is generated but at half of the frequency. For example above, taking bits, 1, 3, 5 and so on, produces the same *PRBS* at half of the frequency. In turn the *PRBS* sequence at one quarter of the frequency can be produced from the half rate *PRBS*, and so on decimating by any factor *R*, where *R* is a power of 2.

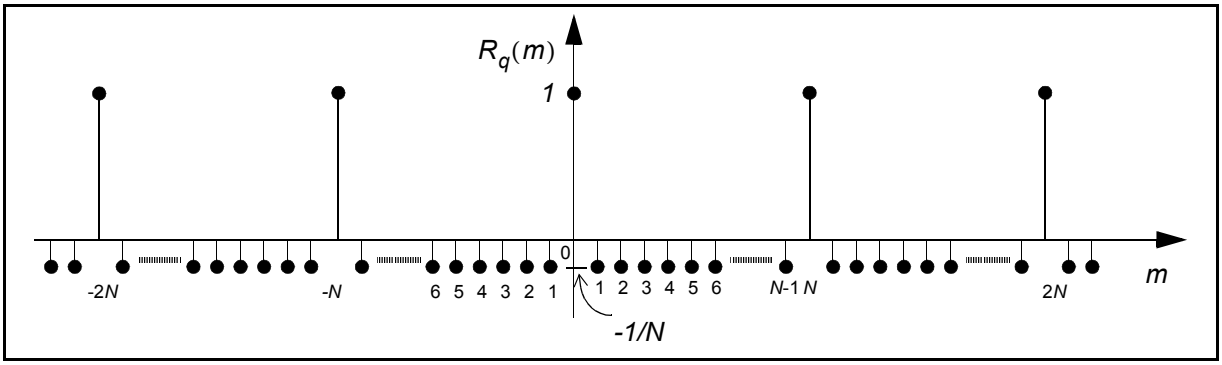
If a signal $q(k)$ is derived from the *PRBS* signal $p(k)$ such that

$$q(k) = \begin{cases} 1 \text{ volt}, & \text{if } p(k) = 1 \\ -1 \text{ volt}, & \text{if } p(k) = 0 \end{cases} \tag{465}$$

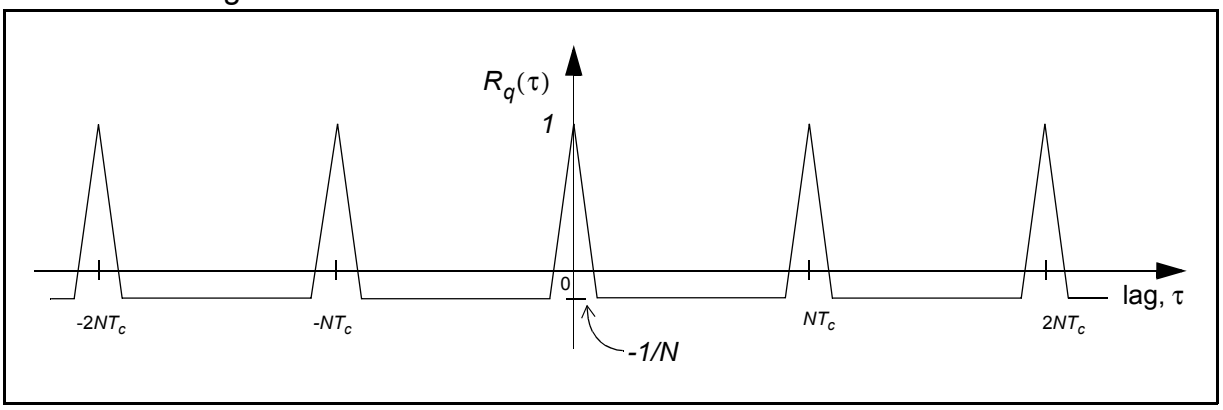
then the autocorrelation of a maximal length *PRBS,* $q(k)$, of *N* bits is:

$$R_q(m) = \frac{1}{L}\sum_{n=0}^{L-1} p(n)p(n+m) = \begin{cases} 1, & m = jN \\ -\dfrac{1}{N}, & m \neq jN \end{cases} \quad \text{where } j = 0, 1, 2, \ldots \tag{466}$$

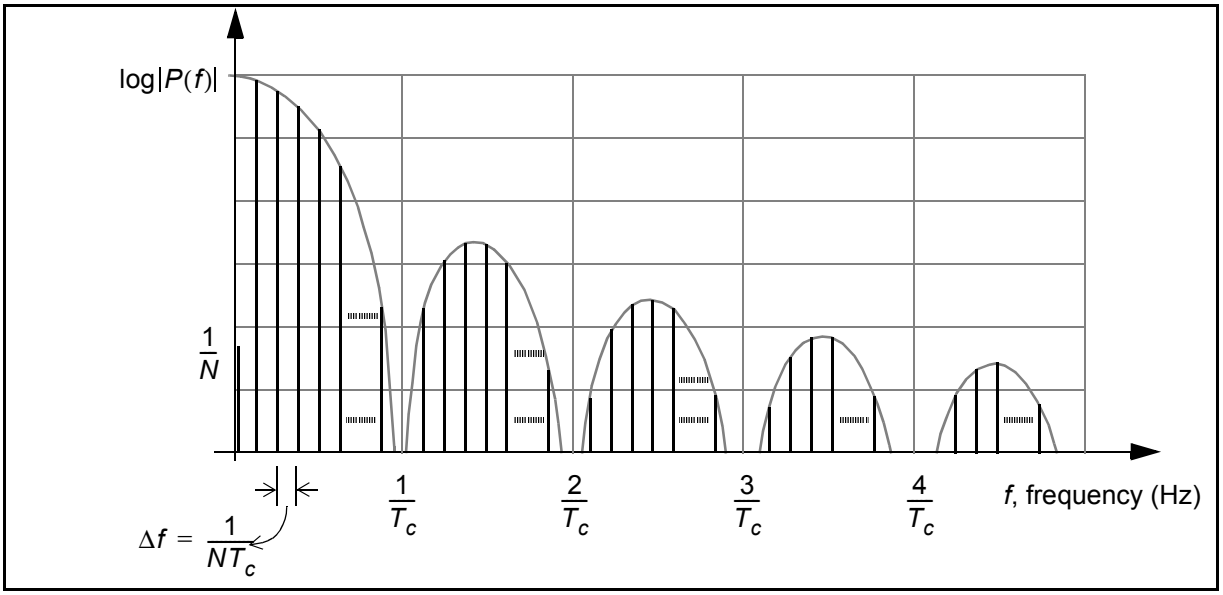
(for *L* large), which can be represented as:



It can therefore be shown that the autocorrelation of the continuous time waveform $q(t)$ is also periodic and is a triangular waveform:



The power spectrum, $P(f)$, obtained from the Fourier transform of the autocorrelation, is therefore a line spectrum, with a $(\sin x/x)^2$ envelope:



Similar types of feedback shift registers to the *PRBS* generator are also used for setting up cyclic redundancy check codes. See also *Characteristic Polynomial, Cyclic Redundancy Check*.

Pseudo-Random Noise Sequence (PRNS): A sequence numbers that has properties that make the sequence appear to be random, in spite of the fact that the numbers are generated in a deterministic way and therefore periodic. Linear feedback shift registers are often used to generate these sequences. Maximal Length (ML) binary sequences produce $2^N - 1$ bit sequences (the longest sequence possible without repetition) from an N bit shift register. See also *Pseudo-Random Binary Sequence*.

Psychoacoustics: The study of how acoustic transmissions are perceived by a human listener. *Psychoacoustics* relates physical quantities such as absolute frequency and sound intensity levels to perceptual qualities, such as pitch, loudness and awareness. Although certain sounds may be presented to the ear, the human hearing mechanism and brain may not perceive these sounds.

For example a simple *psychoacoustic* phenomena is habituation whereby a repetitive sound such as a clock ticking is *not* heard until attention is specifically drawn to it. Spectral masking is an example of a more complex *psychoacoustic* phenomena known whereby loud sounds over a

certain frequency band mask the presence of other quieter sounds with similar frequencies. Spectral masking is now widely exploited to allow data compression of music such as in PASC, Musicam and ATRAC. See also *Adaptive Transform Acoustic Coding, Audiology, Auditory Filters, Beat Frequencies, Binaural Beats, Binaural Unmasking, Equal Loudness Contours, Habituation, Lateralization, Monaural Beats, Precedence Effect, Perceptual Audio Coding, Precision Adaptive Subband Coding (PASC), Sound Pressure Level, Sound Pressure Level Weighting Curves, Spectral Masking, Temporal Masking, Temporary Threshold Shift, Threshold of Hearing.*

Psychoacoustic Model: A model of the human hearing mechanism based on aspects of the human perception of different sounds to the actual sounds being played. For example a *psychoacoustic model* for the phenomenon known as spectral masking has been realized and used to facilitate data compression technique for digital compact cassette (DCC), and for the mini-disc (MD). See also *Psychoacoustics, Precision Adaptive Subband Coding (PASC), Spectral Masking, Temporal Masking, Threshold of Hearing.*

Ptolemy: An object oriented framework for discrete event simulation and DSP systems, design, testing and simulation. *Ptolemy* is available from University of Berkeley.

Pulse Amplitude Modulation (PAM): PAM is a term generally used to refer to communication via a sequence of analog values such as would be needed to send the voltages corresponding to a sampled *but not quantized* analog signal. When the set of values the samples can take on is finite, the term Amplitude Shift Keying (ASK) is usually used to denote this *digital* modulation technique. However, PAM is sometimes used interchangeably with ASK. See also *Sampling, Amplitude Shift Keying.*

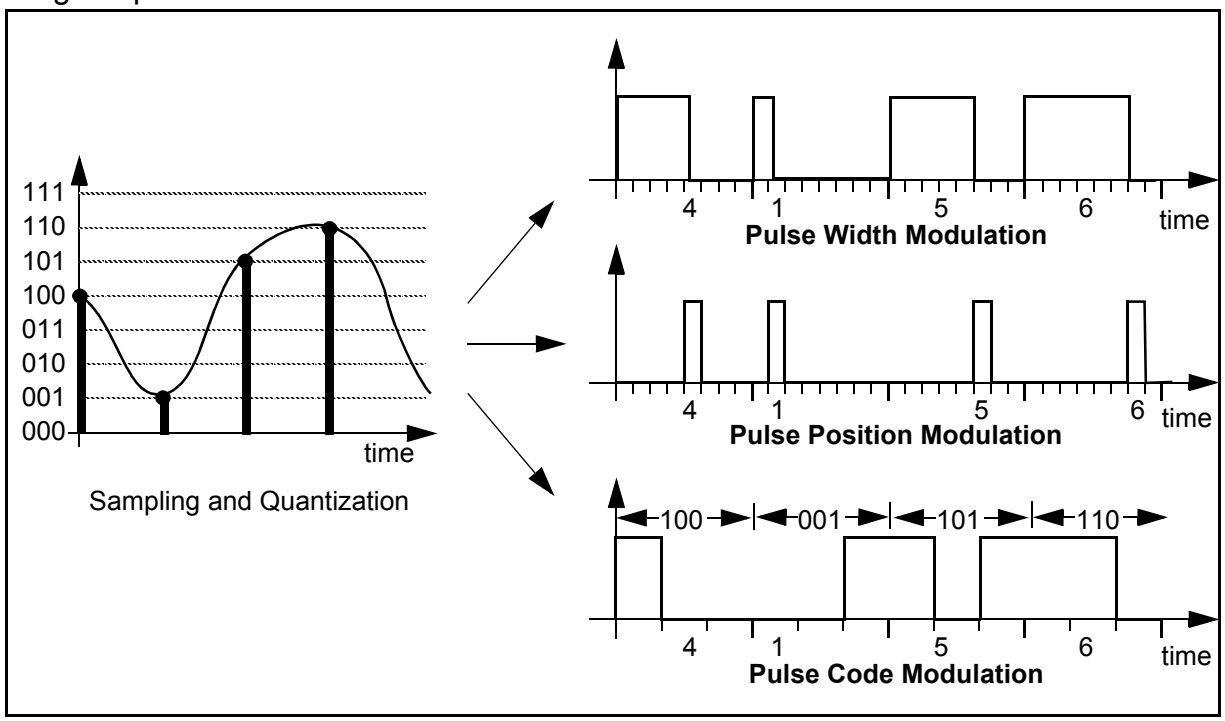
Pulse Code Modulation (PCM): If an analog waveform is sampled at a suitable frequency, then each sample can be quantized to a value represented by a binary code (often 2's complement). The number of bits in the binary code defines the voltage quantization level, and the sampling rate should be at least twice the maximum frequency component of the signal (the Nyquist rate). See also *Analog to Digital Converter, Digital to Analog Converter*. See figure after *Pulse Width Modulation*.

Pulse Position Modulation (PPM): If an analog waveform is sampled at a suitable frequency then the value of each sample can be represented by a single pulse that has a variable position within the sample period that is proportional to the sample analog value. Signals that are received in PPM can be converted back to analog by comparing the samples with a sawtooth waveform. When the pulse is detected, the level of the sawtooth at that time represents the analog value. The earlier a pulse is detected, the lower the analog value. See figure after *Pulse Width Modulation*.

Pulse Train: A periodic train of single unit pulses. *Pulse trains* with a period equalling human voice pitch are used as excitation in vocoding (voice coding) schemes such as linear predictive coding (LPC). See *Linear Predictive Coding, Square Wave*.

Pulse Width Modulation (PWM): PWM is similar to Pulse Position Modulation except that is the information is coded as a the width of a pulse rather than its position in the symbol period. The pulse

width is proportional to the analog value of that sample. The analog signal can be recovered by integrating the pulses.



Pythagorean (Music) Scale: Prior to the existence of the equitemporal or Western music scale, a (major) musical key was formed from using certain carefully chosen frequency ratios between adjacent notes, rather than the constant tone and semitone ratios of the modern Western music scale. The ancient C-major *Pythagorean scale* would have had the following frequency ratios:

C-major Scale	**C**	**D**	**E**	**F**	**G**	**A**	**B**	**C**
Frequency ratio	**1/1**	**9/8**	**81/64**	**4/3**	**3/2**	**27/16**	**243/128**	**2/1**

The frequency ratio gives the ratio of the fundamental frequency of the root note, to the current note. The above ratios correspond to the *Pythagorean Music Scale*.

Any note can be used to realise a Pythagorean major key or scale. However using the *Pythagorean scale* it is difficult to form other major or minor keys without a complete retuning of the instrument. Instruments that are tuned and played using the *Pythagorean scale* will probably sound in some sense “ancient” as our modern appreciation of music is now firmly based on the equitempered Western music scale. See also *Digital Audio, Just Scale, Music, Music Synthesis, Western Music Scale..*

Q

Q Format: Representing binary numbers in the *Q format* ensures that all numbers have a magnitude between **-**1 and 1. The MSB of a Q15 number is the sign bit with magnitude 1, and the bits following have bit values of:

$2^{-1} = 0.5$, $2^{-2} = 0.25$, $2^{-3} = 0.125$,..... $2^{-15} = 3.0517578 \times 10^{-5}$.

The only difference between normal two's complement (binary point after the LSB) and Q format is the position of the binary point.

The Q format is used in DSP to ensure that when two numbers are multiplied together their magnitude will always be less than 1. Therefore fixed point DSP processors can perform arithmetic without overflow.

QR: See *Matrix Decompositions - QR*.

QR Algorithm: A linear technique that implicitly forms an orthogonal matrix **Q** to transform a matrix **A** into an upper triangular matrix **R**, i.e. **A = QR**. The *QR* algorithm is numerically stable and can be used for solving linear sets of equations in a variety of DSP applications from speech recognition to beamforming. The algorithm is however, very computationally expensive and not used very often for real time DSP. See *Matrix Decompositions - QR.*

Quad: A prefix to mean "four of". For example the Burr Brown DAC4814 chip is described as a *Quad* 12 Bit Digital to Analog Converter (DAC) meaning that the chip has four separate (or independent) DACs. See also *Dual*.

Quadraphonic (or Quadrophonic): Using four independent channels for the reproduction of hi-fidelity music. Quadrophonic systems were first introduced in the 1970s as an enhancement to the stereophonic system, however the success was limited. In the 1990s surround sound systems such as Dolby Prologic use four and more channels to encode the sound with 3-dimensional effect. The term quadraphonic is rarely implemented or used. Note that a system which simply uses four loudspeakers (two left channels and two right channels) is not quadraphonic. See also *Stereophonic, Surround Sound, Dolby Prologic*.

Quadratic Equation: A polynomial is a *quadratic equation* if it has the form, $ax^2 + bx + c = 0$, where x is a variable, and a,b, and c are constants. Note that the quantity x may be a vector, and a, b, and c are appropriately dimensioned vectors and matrices. For example in calculating the Wiener-Hopf solution the following equation must be solved:

$$\boldsymbol{x}\boldsymbol{R}\boldsymbol{x}^T + \boldsymbol{p}\boldsymbol{x} + c = \boldsymbol{0} \tag{467}$$

where $\boldsymbol{x}$ is an $n \times 1$ vector, $\boldsymbol{R}$ is an $n \times n$ matrix, $\boldsymbol{p}$ is an $n \times 1$ vector and c is a scalar constant.

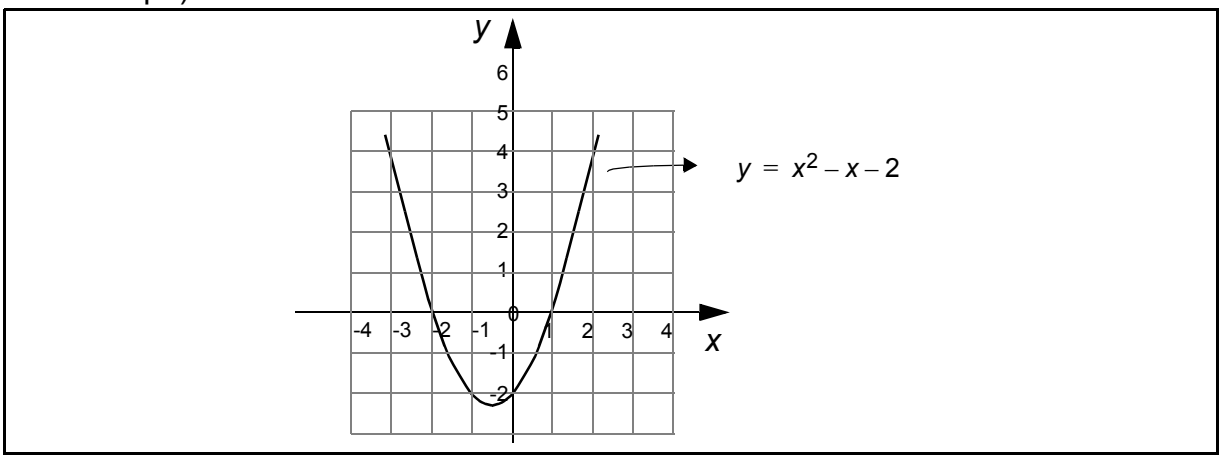
Quadratic Formula: Given a quadratic polynomial, $ax^2 + bx + c = 0$, the roots of this polynomial can be calculated from:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{468}$$

such that:

$$\left(x + \frac{b+\sqrt{b^2-4ac}}{2a}\right)\left(x + \frac{b-\sqrt{b^2-4ac}}{2a}\right) = x^2 + \frac{b}{a}x + \frac{c}{a} \tag{469}$$

Geometrically, the roots of a polynomial are where a graph of $y = ax^2 + bx + c$ (which is parabolic in shape) cuts the x-axis.



Note that if the graph does not cut the x-axis, then the quantity $\sqrt{b-4ac}$ will be an imaginary number (square root of a negative number), and the roots are then complex numbers. See also *Complex Roots, Poles, Polynomial, Zeroes*.

Quadratic Surface: See *Hyperparaboloid.*

Quadrature: This term is used in reference to the four quadrants defined in two dimensions. *Quadrature* representations are particularly useful in communications because the cosine and sine components of a single frequency can be thought of as the two axes in the complex plane. By representing signals via in-phase (cosine) and quadrature (sine) components, all of the tools of complex number analysis are available to simplify the analysis and design of digital signal sets.

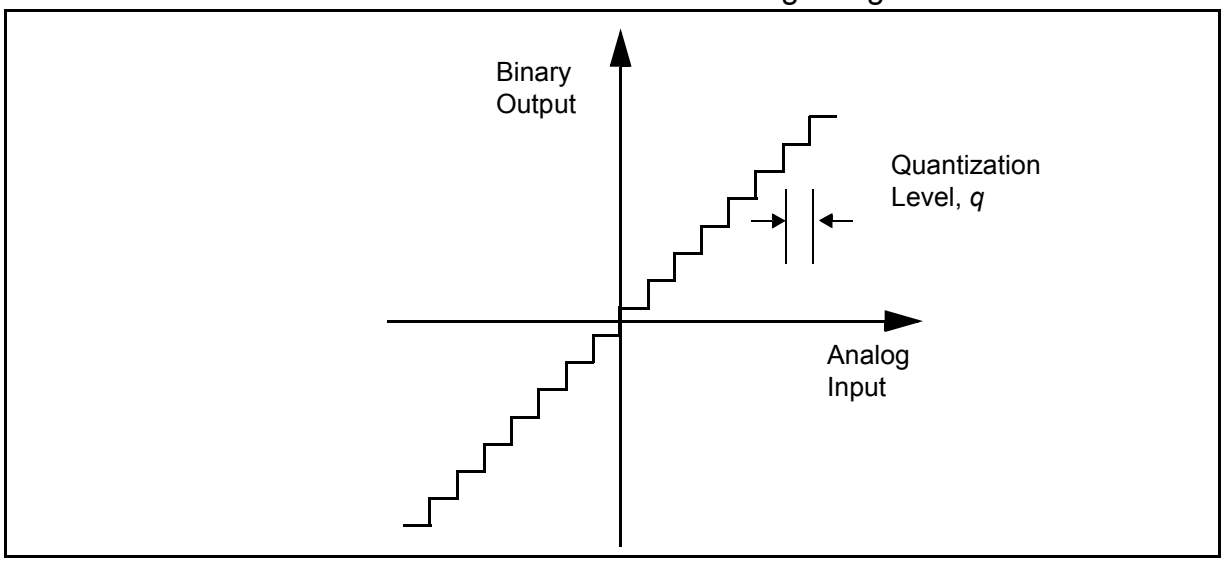
Quadrature Amplitude Modulation (QAM): When both the amplitude and the phase of a quadrature (two dimensional) signal set are varied to encode the information bits in a digital communication system, the modulation technique is often referred to as QAM. Common examples are rectangular signal sets defined on a two-dimensional Cartesian lattice, such as 16 QAM (4 bits per symbol), 32 QAM (5 bits per symbol), and 64 QAM (6 bits per symbol). QAM modulation techniques are used for many modem communication standards. See also *V-Series Recommendations, Amplitude Shift Keying, Phase Shift Keying.*

Quadrature Mirror Filters (QMF): A type of digital filter which has special properties making it suitable for sub-band coding filters.

Quadrature Phase Shift Keying (QPSK): QPSK is a common digital modulation (phase shift keying) technique that uses four signals (symbols) that have equal amplitude and are successively shifted by 90 degrees in phase. See also *Phase Shift Keying, Quadrature.*

Quantization: Converting from a continuous value into a series of discrete levels. For example, a real value can be quantized to its nearest integer value (rounding) and the resulting error is referred

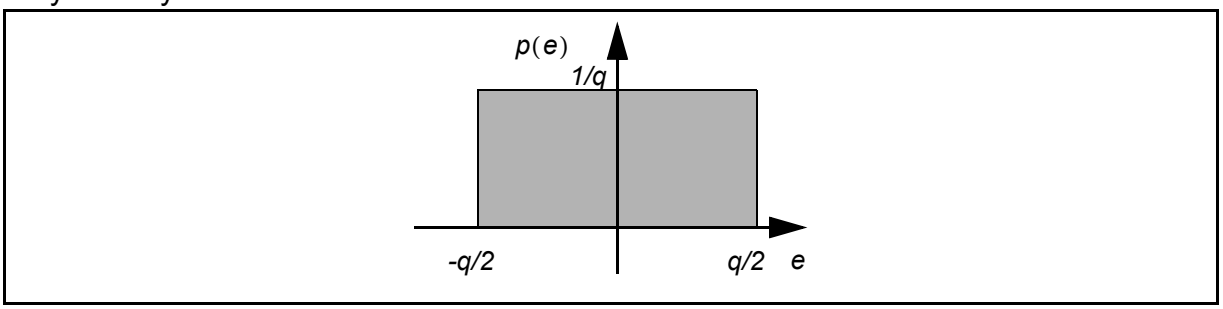
to as the quantization error. The quantization error therefore reflects the accuracy of an ADC. Quantization introduces an irreversible distortion on an analogue signal.



Quantizers are found somewhere at the heart of every lossy compression algorithm. In JPEG, for example, the quantizer appears when the DCT coefficients for an image block are quantized. See also *Analog to Digital Converter, A-law C, Sample and Hold.*

Quantization Error: The difference between the true value of a signal and the discrete value from the A/D at that particular sampling instant. If the quantization level is *q* volts, then the maximum error at each sample is *q*/2 volts. If an analog value *x* is to be quantized it is convenient to represent the quantized value as a sum of the true analog value and a *quantization error* component, *e*, i.e.: $\bar{x} = x + e$, where $\bar{x}$ is the quantized value of *x.* See also *Rounding Noise, Truncation Noise.*

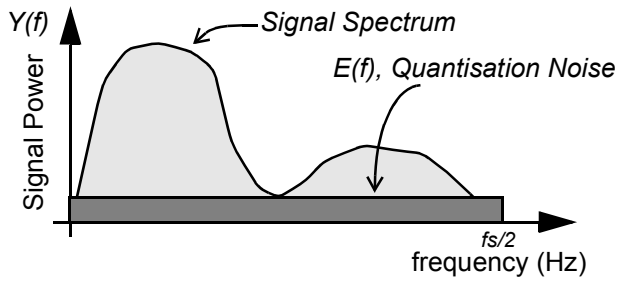
Quantization Noise: Assuming the an ADC rounds to the nearest digital level, the maximum quantisation error of any one sample is *q/2* volts (see *Quantization Error*). If we assume that the probability of the error being at a particular value between *+q/2* and *-q/2* is equally likely then the probability density function for the error is flat.



Therefore treating the error as *white noise*, then we can calculate the noise power of the error as:

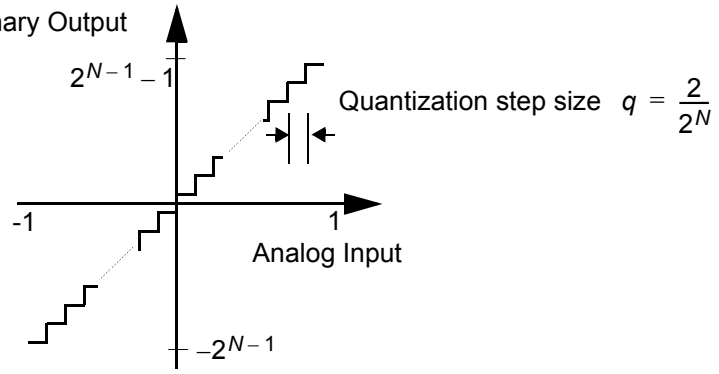
$$n_{adc} = \frac{1}{q}\int_{-q/2}^{q/2} e^2 de = \frac{q^2}{12} \qquad (470)$$

The quantisation noise will extend over the frequency range 0 to $f_s/2$. , i.e. the full baseband.



Low level signals may be masked by the quantisation noise. Although it is assumed that the quantisation noise is uncorrelated with the signal, in practice for periodic signals this is not strictly true, and therefore the flat white spectrum is not strictly true.

For an *N*-bit signal, there are 2^N levels from the maximum to the minimum value of the quantiser:



Therefore the mean square value of the quantisation noise power can be calculated as:

$$Q_N = 10\log\left(\frac{(2/2^N)^2}{12}\right) = 10\log 2^{-2N} + 10\log\frac{4}{12} \approx -6.02N - 4.77 \text{ dB} \quad (471)$$

Another useful measurement is the signal to quantisation noise ratio (SQNR). For the above ADC with voltage input levels between -1 and +1 volts, if the input signal is the maximum possible, i.e. a sine wave of amplitude 1 volt, then the average input signal power is:

$$\text{Signal Power} = E[\sin 2\pi ft^2] = \frac{1}{2} \quad (472)$$

Therefore the maximum SQNR is:

$$\text{SQNR} = 10\log\frac{\text{Signal Power}}{\text{Noise Power}} = 10\log\frac{0.5}{\left(\frac{(2/2^N)^2}{12}\right)} = 10\log 2^{-2N} + 10\log\frac{3}{2} \quad (473)$$

$$= 6.02N + 1.76 \text{ dB}$$

For a *perfect* 16 bit ADC the quantisation noise can be calcuated to be 98.08 dB. See also *A-law compression, Signal to Noise Ratio*.

Quantisation Noise, Reduction by Oversampling: Oversampling can be used to increase the resolution of an ADC or DAC. If an ADC has a step size of q volts (see *Quantisation Error*) and a Nyquist sampling rate of f_n, then the maximum error, $e(n)$, of a quantised sample is between $-q/2$ and $q/2$. Therefore if the true sample value is $x(n)$, then the quantised sample, $y(n)$, is:

$$y(n) = x(n) + e(n) \tag{474}$$

If we assume that the quantisation value is equally likely to take any value in this range (i.e. it is white), then we can assume that the probability density function for the noise signal is uniform. Therefore the average quantisation noise power in the range 0 to $f_n/2$ can be calculated as the average squared value of e:

$$Q_N = \frac{1}{q}\int_{-q/2}^{q/2} e^2 de = \frac{1}{q}\frac{1}{3}e^3\Big|_{-q/2}^{q/2} = \frac{q^2}{12} \tag{475}$$

The same answer could be obtained from the time average:

$$Q_N \approx \frac{1}{M}\sum_{m=0}^{M-1} e^2(m) = \frac{q^2}{12} \tag{476}$$

In order to appreciate that the quantisation noise does not decrease, note that the same approximate answer is obtained for a signal that is oversampled by R times:

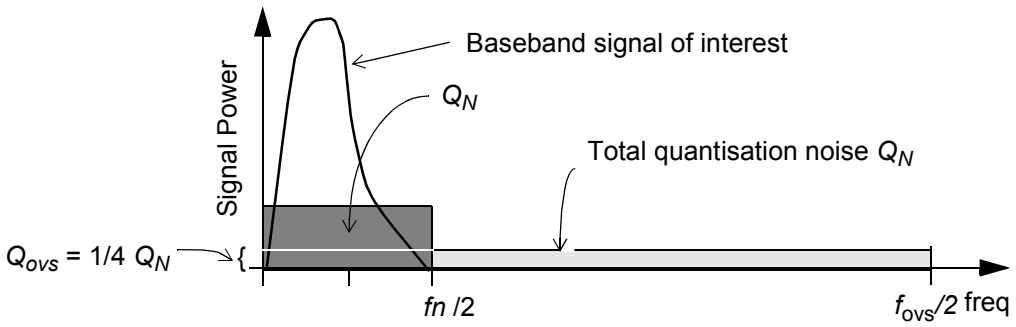
$$Q_N \approx \frac{1}{MR}\sum_{r=0}^{MR-1} e^2(r) = \frac{q^2}{12} \tag{477}$$

For an oversampled system sampling at f_{ovs} and using the same converter, the total quantisation noise power will of course be the same but because it is white (a flat spectrum) it is now spread over the range 0 to $f_{\text{ovs}}/2$. Evaluating Eqs. 475 or 476 for different sampling rates will give the same answer. The actual noise power in the baseband, Q_{ovs}, is now given as:

$$Q_{ovs} = \frac{q^2(f_n/2)}{12(f_{\text{ovs}}/2)} \tag{478}$$

(Note that for the more common periodic and aperiodic signals, the quantisation noise spectra is not “white”; however for a “noisy” stochastic input signal the white quantisation noise assumption is “reasonably” valid). From Eq. 478, in order to increase the baseband signal to quantisation noise ratio we can either increase the number of bits in the ADC or increase the sampling rate above the Nyquist rate. By increasing the sampling rate, the total quantisation noise power does not increase, and as a result the in-band quantisation noise power will decrease.

As an example, oversampling a signal by a factor of 4 x's the Nyquist rate reduces the in-band quantisation noise by 1/4:



When a signal is oversampled the total level of quantisation noise does not change. Therefore for every increase in sampling rate above Nyquist the baseband quantisation noise power will reduce.

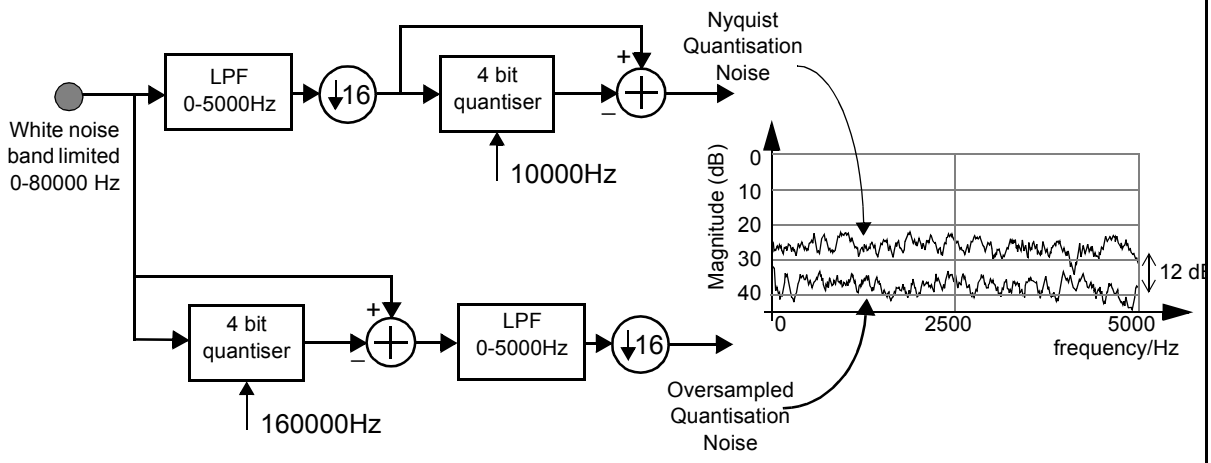
This level of baseband noise power is equivalent to an ADC with step size $q/2$:

$$Q_{\text{OVS}} = \frac{q^2(f_n/2)}{12(4f_n/2)} = \frac{2(q/2)^2}{12} = \frac{Q_N}{4} \tag{479}$$

and hence baseband signal resolution has been increased by 1 bit since. For each extra bit of resolution the signal to quantisation noise ratio improves by $20\log 2 = 6.02$ dB. In theory therefore if a single bit ADC were used and oversampled by a factor of 4^{15} ($\approx 10^9 \times f_n$) then a 16 bit resolution signal could be realized! Clearly this sampling rate is not practically realisable. However on a more pragmatic level, if a well trimmed low noise floor 8 bit ADC converter was used to oversample a signal by a factor of 16 x's the Nyquist rate, then when using a digital low pass filter to decimate the signal to the Nyquist rate, approximately 10 bits of resolution could be obtained. Single bit oversampling ADCs can however still be achieved using quantisation noise shaping strategies within sigma delta converters (see *Sigma Delta*).

To illustrate increasing the signal resolution by oversampling, the figure below shows the result of a simulation quantising a high resolution floating point white noise digital signal in the amplitude range -1 to +1 to 4 bits (i.e. 16 levels in range -1 to +1) using a digital quantiser to simulate an ADC. The bandwidth of interest is 0-5000 Hz, and hence the Nyquist rate is $f_n = 10000$ Hz, and oversampling at 16 x's gives $f_{\text{ovs}} = 160000$ Hz and should yield two "extra bits" of resolution. The quantisation noise for the Nyquist rate and oversampled rate quantisers (ADCs) then reveals the

expected 12 dB advantage from the oversampling strategy.



Quantising a real value (floating point) signal of baseband 0-5000 Hz to 4 bits. Note that the oversampling procedure produces a level of inband quantisation noise that is 12 dB below that of the Nyquist rate quantiser. The magnitude spectra was produced from a 1024 point FFT of the quantisation noise, and smoothed by a window of length 8. The input white noise signal was 16384 samples.

See also *Decimation, Interpolation, Oversampling, Quantization, Sigma Delta Converter*.

Quarter Common Intermediate Format (QCIF): The *QCIF* image format is 144 lines by 180 pixels/line of luminance and 72 x 90 of chrominance information and is used in the |TU-T H.261 digital video recommendation. A full version of QCIF called CIF (common image format) is also defined in H.261. The choice between *CIF* or *QCIF* depends on available channel capacity and desired quality. See also *Common Intermediate Format*, *H-series Recommendations, International Telecommunication Union*.

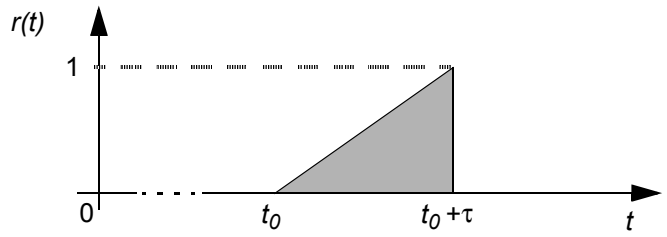
Quicksilver: A versatile, if difficult to find, software package.

Quicktime: A proprietary algorithm for video compression using very low levels of processing to allow real time implementation in software on and Macintosh computers [79]. Quicktime does not achieve the picture quality of techniques such as *MPEG1*. See also *MPEG1.*

R

Ramp Waveform (Continuous and Discrete Time): The ***continuous ramp waveform*** can be defined as:

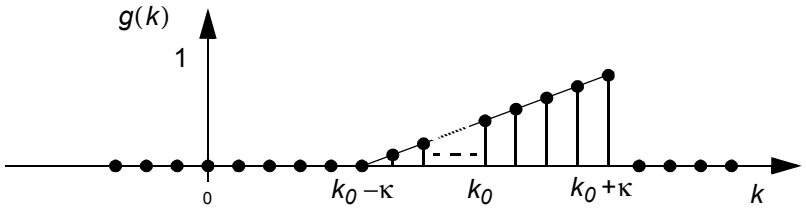
$$\text{ramp}((t-t_0)/\tau) = \begin{cases} \dfrac{t-t_0}{\tau} & \text{if } 0 \le (t-t_0) < \tau \\ 0 & \text{otherwise} \end{cases} \qquad \text{continuous time} \tag{480}$$



The *continuous triangular pulse* $r(t) = \text{ramp}((t-t_0)/\tau)$

The ***discrete time ramp waveform*** can be defined as:

$$\text{ramp}((k-k_0)/\kappa) = \begin{cases} \dfrac{k-k_0}{\kappa} & \text{if } 0 \le (k-k_0) < \kappa \\ 0 & \text{otherwise} \end{cases} \qquad \text{discrete time} \tag{481}$$



$g(k) = \text{tri}((k-k_0)/\kappa)$

See also *Elementary Signals, Rectangular Pulse, Sawtooth Waveform, Square Wave, Triangular Pulse, Unit Impulse Function, Unit Step Function*.

Random Access Memory (RAM): Digital memory which can be used to read or write binary data to. *RAM memory* is usually volatile, meaning that it loses information when the power is switched off. Non-volatile *RAM* is available. See also *Non-Volatile, Static RAM, Dynamic RAM.*

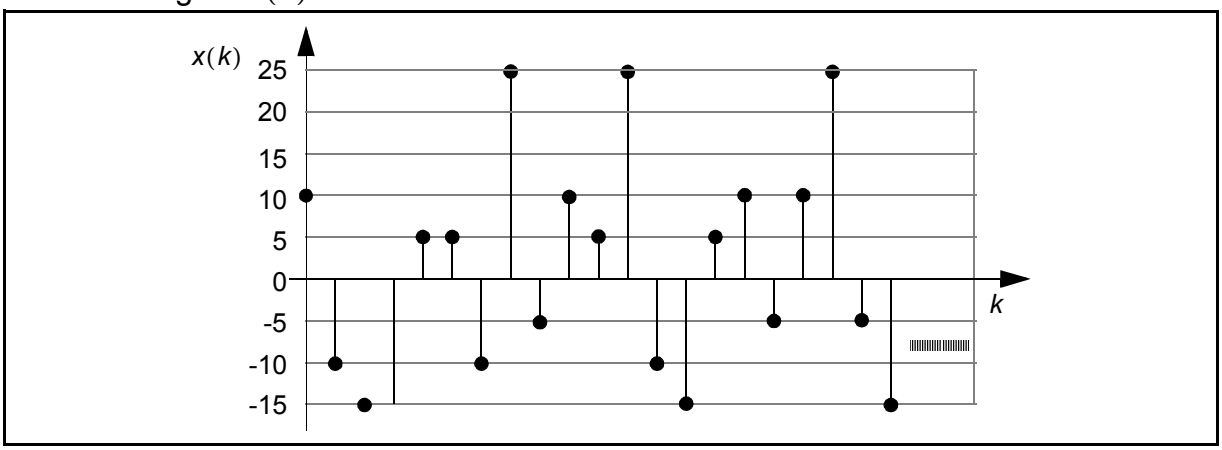
Random Variable: A *random variable* is a real valued function which is defined based on the outcomes of a probabilistic system. For example a die can be used to create a signal based on the *random variable* of the die outcome. The probabilistic event is the shaking of the die where each

independent event is denoted by k, and there are 6 equally likely outcomes. A particular *random variable* $x(k)$ can be defined by the following table:

Die Event	Random Variable $x(.)$	$p(x(.))$
1	-15	1/6
2	-10	1/6
3	-5	1/6
4	+5	1/6
5	+10	1/6
6	+25	1/6

Table 1:

and the random signal $x(k)$ turns out to be:



The time average of the signal $x(k)$, denoted as $\bar{x}$, can be calculated as:

$$\bar{x} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N} x(k) = 1.6666\ldots \tag{482}$$

The statistical mean, and denoted as $E[x(k)]$, where $E[.]$ is the expectation operator can be calculated as:

$$\begin{aligned} E[x(k)] &= \sum_{x} p(x)x, \text{ for all values of } x \\ &= \left(25\cdot\frac{1}{6}\right)+\left(10\cdot\frac{1}{6}\right)+\left(5\cdot\frac{1}{6}\right)-\left(5\cdot\frac{1}{6}\right)-\left(10\cdot\frac{1}{6}\right)-\left(15\cdot\frac{1}{6}\right) \\ &= 1.6666\ldots \end{aligned} \tag{483}$$

The time average mean squared value, denoted as $\overline{x^2}$, can be calculated as:

$$\overline{x^2} = \lim_{N\to\infty}\frac{1}{N}\sum_{k=0}^{N} x^2(k) = 183.333\ldots \tag{484}$$

The statistical average squared value, denoted as $E[x^2(k)]$ can be calculated from:

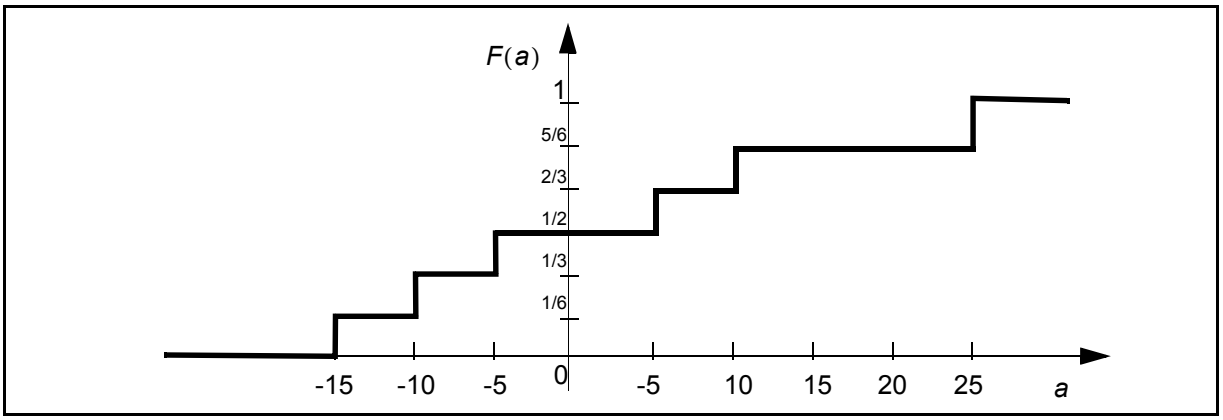
$$\begin{aligned} E[x^2(k)] &= \sum_x p(x)x^2, \text{ for all values of } x \\ &= \left(625\cdot\frac{1}{6}\right)+\left(100\cdot\frac{1}{6}\right)+\left(25\cdot\frac{1}{6}\right)-\left(25\cdot\frac{1}{6}\right)-\left(100\cdot\frac{1}{6}\right)-\left(225\cdot\frac{1}{6}\right) \\ &= 183.333\ldots \end{aligned} \tag{485}$$

If the random process generating $x(k)$ is ergodic, then the statistical averages equal the time averages, i.e. $\bar{x} = E[x(k)]$ and $\overline{x^2} = E[x^2(k)]$.

For a particular *random variable*, $x(k)$, a cumulative distribution function can be specified, where:

$$F(a) = P(x(k)\le a) = \lim_{n\to\infty}\left(\frac{\text{no. of values of } x(k)\le a}{\text{total no. of values, } n}\right) \tag{486}$$

i.e., $F(a)$ specifies the probability that the value $x(k)$ is less than *a*. Therefore for the above *random variable*, $x(k)$, the cumulative distribution function is:



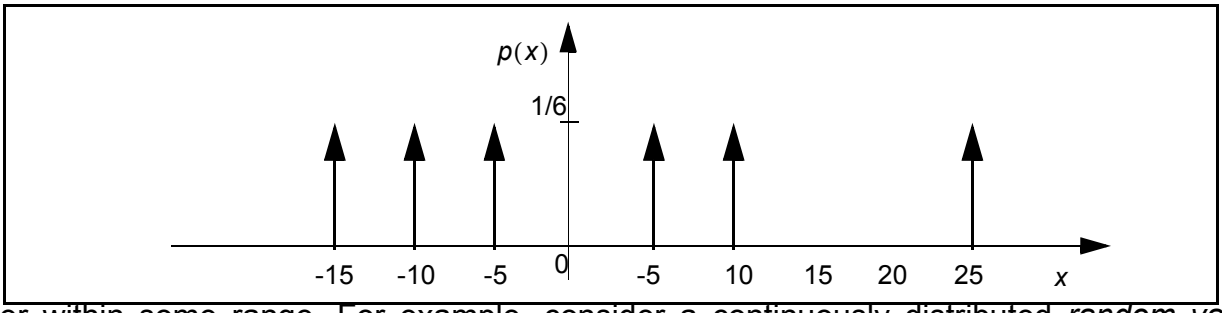
The probability density function (PDF) is defined as:

$$p(x) = \left.\frac{dF(a)}{dx}\right|_{a=x} = \left.\frac{dP(x\le a)}{dx}\right|_{a=x} \tag{487}$$

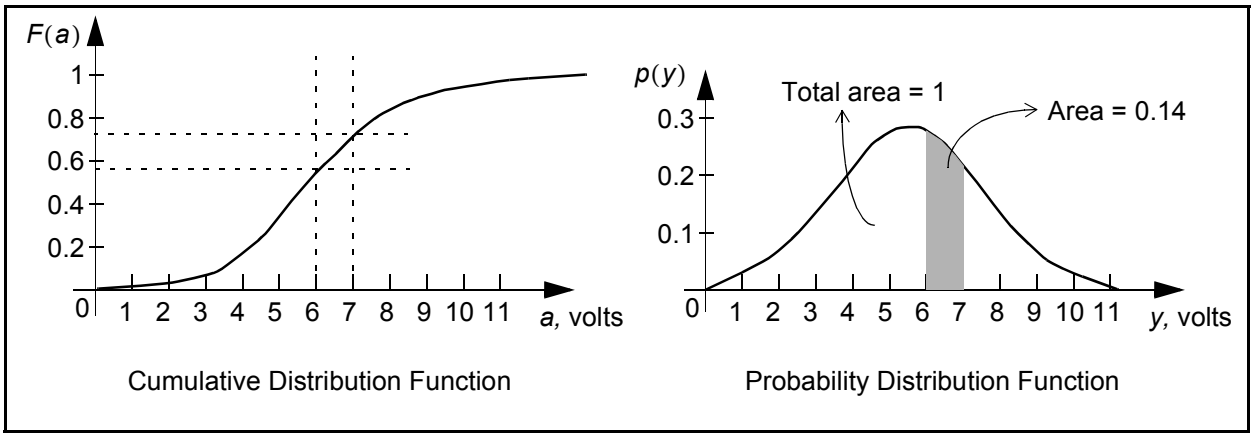
where the "(k)"has been dropped for notational convenience. The PDF for the *random variable* $x(k)$ produced by the probabilistic events of a die shake is therefore:

where the arrows represent dirac-delta functions located at the discrete values of the *random variable*. Therefore the total area under the graph $p(x)$ is 1.

The above distributions are discrete, in that the *random variable* can only take on specific values and therefore the distribution function increases in steps, and the PDF consists of dirac delta functions. There also exist continuous distributions where the *random variable* can take on any real



number within some range. For example, consider a continuously distributed *random variable* which denotes the *exact* voltage measured from a 9 volt battery. By measuring the voltage of a large number of batteries, a *random variable* $y(.)$ denoting the battery voltages can be produced. For a particular batch of a few thousand batteries the distribution function and PDF obtained are:



If, for example, it is required to calculate the probability of a battery having a voltage between 6 and 7 volts, then the area under the PDF between y values of 6 and 7 can be calculated, or the appropriate values of the distribution function subtracted:

$$P(6 < y \leq 7) = \int_6^7 p(y)dy = F(7) - F(6) = 0.14 \qquad (488)$$

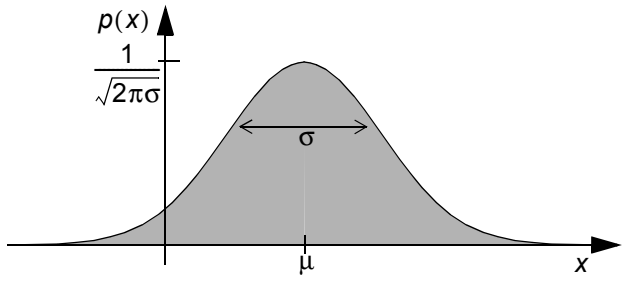
In DSP signals with both discrete and continuous distributions are found. For example thermal noise is continuously distributed signal, whereas the sequence of character symbols typically sent by a modem has a discrete distribution.

Some important discrete distributions in DSP are:

- Binomial;
- Poisson;

Some important continuous probability density functions in DSP are:

• **Gaussian:**

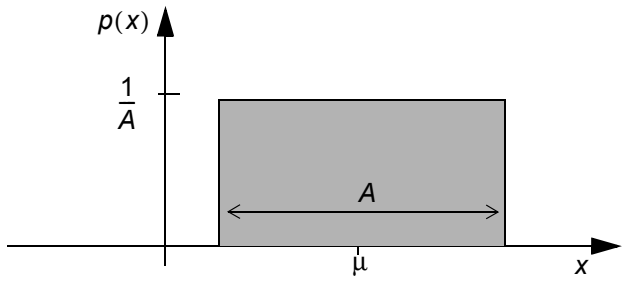


$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Mean, $E[x] = \mu$

Variance, $E[(x-\mu)^2] = \sigma^2$

• **Uniform:**



$$p(x) = \begin{cases} 0, x < (2\mu - A)/2 \\ 1/A, |x - m| \le A/2 \\ 0, x > (2\mu - A)/2 \end{cases}$$

Mean, $E[x] = \mu$

Variance, $E[(x-\mu)^2] = \frac{A^2}{12}$

The n-th moment of a PDF taken about the point $x = x_0$ is:

$$E[(x - x_o)^n] = \int_{-\infty}^{\infty} (x - x_o)^n p(x) dx \tag{489}$$

The second order moment around the mean, $E[(x - E[x])^2]$ is called the variance or the second central moment.

See also *Ergodic, Expected Value, Mean Value, Mean Squared Value, Probability, Variance, Wide Sense Stationarity.*

Range of Matrix: See *Matrix Properties - Range.*

Rank of Matrix: See *Matrix Properties - Rank.*

Rate Converter: Usually referring to the change of the sampling rate of a signal. See *Decimation, Downsampling, Fractional Sampling Rate Converter, Interpolation, Upsampling.*

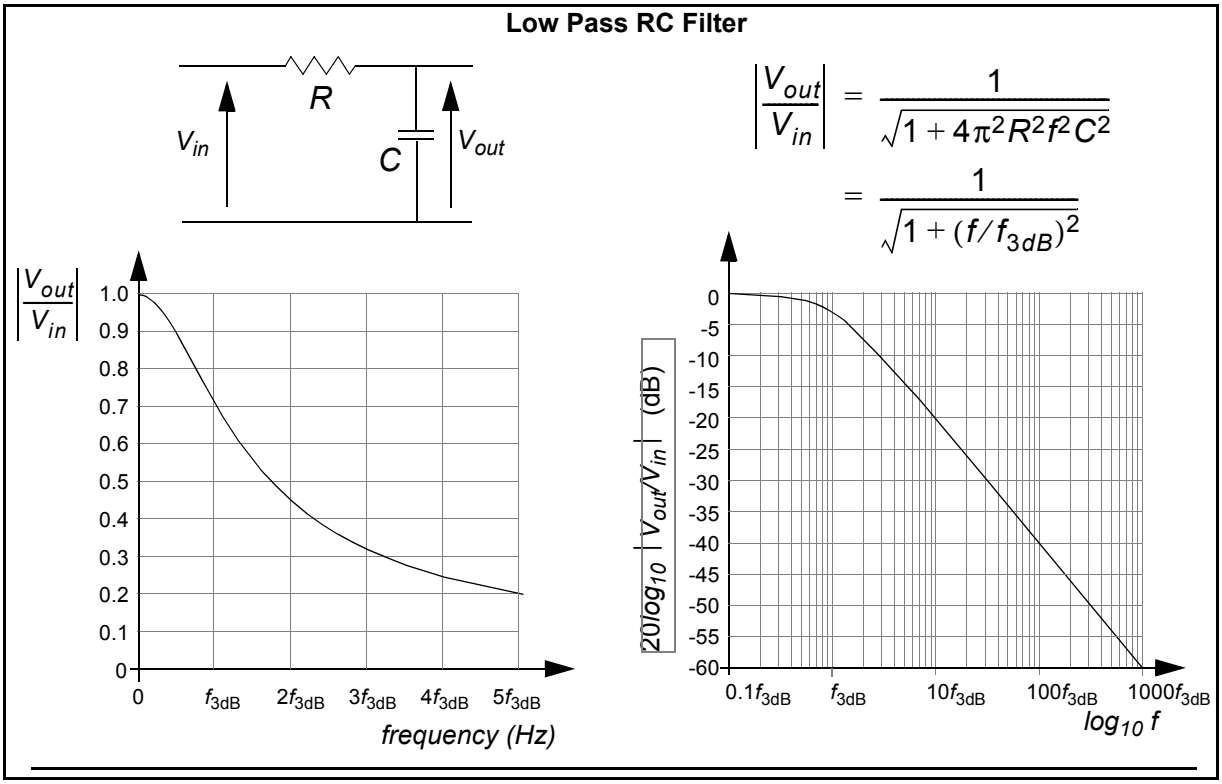
RBDS: An FM data transmission standard that allows radio stations to send traffic bulletins, weather reports, song titles or other information to a display on *RBDS* compatible radios. Radios will therefore be able to scan for a particular type of music. For emergency broadcasting an *RBDS* signal can automatically turn on a radio, turn up the radio volume and issue an emergency alert.

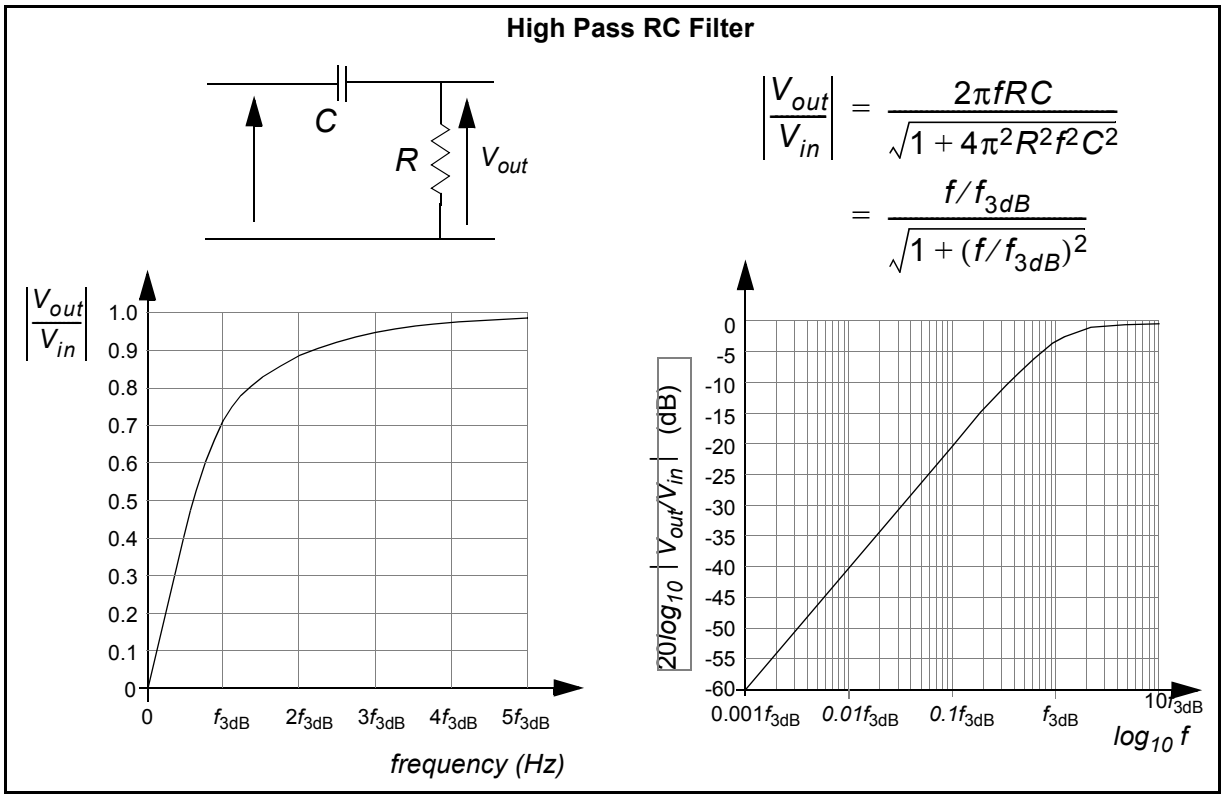
RC Circuit: The very simplest form of analog low pass or high pass filter used in DSP systems. The 3dB point is at $f_{3dB} = 1/(2\pi RC)$. An *RC circuit* is only suitable as a (low pass) anti-alias filter when the sampling frequency is considerably higher than the highest frequency present in the input signal; this is usually only the case for oversampled DSP systems where the anti-alias process is primarily performed digitally. The roll-off for a simple low pass *RC circuit* is 6dB/octave, or 20dB/decade when plotted on a logarithmic frequency scale.

An *RC circuit* can also be used as a differentiator noting that the current through a capacitor is limited by the rate of change of the voltage across the capacitor:

$$i = C\frac{dV}{dt} \tag{490}$$

See also *3dB point, Decade, Differentiator, Logarithmic Frequency, Octave, Oversampling, Roll-off, Sigma Delta*.





Read Only Memory (ROM): Digital memory to which data cannot be written. *ROM* also retains information even when the power is switched off.

Reasoning, Circular: See *Circular Reasoning.*

Real Exponential: See *Exponential, Complex Exponential.*

Real Time Processing: *Real time* is the expression used to indicate that a signal must be processed and output again without any noticeable delay. For example, consider speech being sensed by a microphone before being sampled by a DSP system. Suppose it is required to filter out the low frequencies of the speech before sending the data down a telephone line. The filtering must be done in *real time* otherwise new samples of data will arrive before the system has DSP system has finished its calculations on the previous ones! Systems that do not operate in real time are often referred to as off-line. See also *Off-Line Processing.*

Reciprocal Polynomial: Consider the polynomial:

$$H(z) = a_1 + a_2 z^{-1} + \ldots + a_{N-1} z^{-N+1} + a_N z^{-N} \tag{491}$$

The *reciprocal polynomial* is given by:

$$H_r(z) = a_N^* + a_{N-1}^* z^{-1} + \ldots + a_1^* z^{-N+1} + a_0^* z^{-N} \tag{492}$$

where a_i^* is the complex conjugate of a_i. The polynomials are so called because the reciprocals of the zeroes of $H(z)$ are the zeroes of $H_r(z)$. If $H(z)$ factorises to:

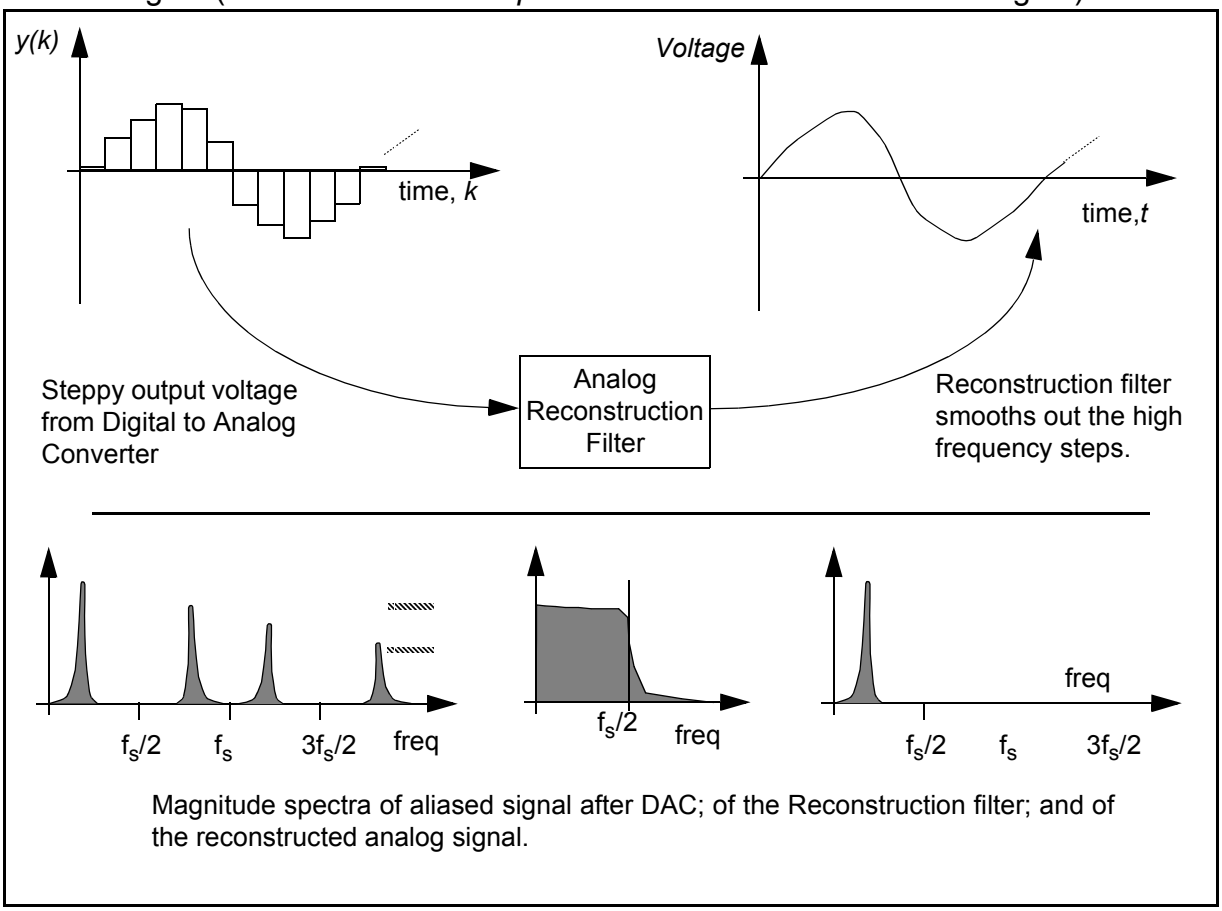
$$H(z) = (1-\alpha_1 z^{-1})(1-\alpha_2 z^{-1})\ldots(1-\alpha_{N-1} z^{-1})(1-\alpha_N z^{-1}) \tag{493}$$

then the zeroes of the order reversed polynomial are $\alpha_1^{-1}, \alpha_2^{-2}, \ldots \alpha_{N-1}^{-1}, \alpha_N^{-1}$ which can be seen from:

$$\begin{aligned} H_r(z) &= z^{-N}H(z^{-1}) \\ &= z^{-N}(1-\alpha_1 z)(1-\alpha_2 z)\ldots(1-\alpha_{N-1}z)(1-\alpha_N z) \\ &= (z^{-1}-\alpha_1)(z^{-1}-\alpha_2)\ldots(z^{-1}-\alpha_{N-1})(z^{-1}-\alpha_N) \\ &= \frac{(-1)^N}{\alpha_1\alpha_2\ldots\alpha_{N-1}\alpha_N}(1-\alpha_1^{-1}z^{-1})(1-\alpha_2^{-1}z^{-1})\ldots(1-\alpha_{N-1}^{-1}z^{-1})(1-\alpha_N^{-1}z^{-1}) \end{aligned} \quad (494)$$

Reciprocal polynomials are of particular relevance to the design of all pass filters. See *All-pass Filter, Finite Impulse Response, Order Reversed Filter.*

Reconstruction Filter: The analog filter at the output of a DAC to remove the high frequencies present in the signal (in the form of the *steps* between the discrete levels of signal).

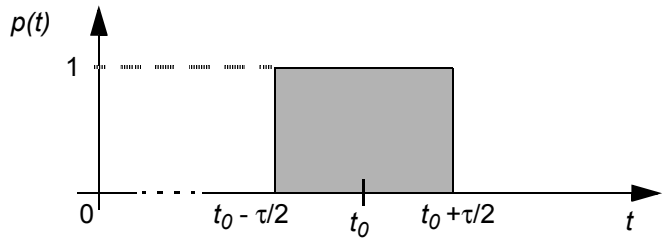


Magnitude spectra of aliased signal after DAC; of the Reconstruction filter; and of the reconstructed analog signal.

Rectangular Matrix: See *Matrix Structured - Rectangular.*

Rectangular Pulse (Continuous and Discrete Time): The ***continuous time rectangular pulse*** can be defined as:

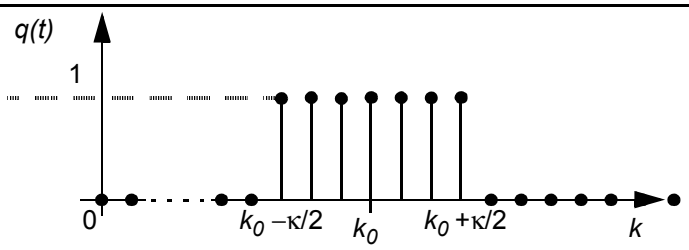
$$\text{rect}((t-t_0)/\tau) = \begin{cases} 1 \text{ if } |t-t_0| < \tau/2 \\ 0 \text{ otherwise} \end{cases} \quad \text{continuous time} \tag{495}$$



The *continuous rectangular pulse* $p(t) = \text{rect}((t-t_0)/\tau)$

The ***discrete time rectangular pulse*** can be defined as:

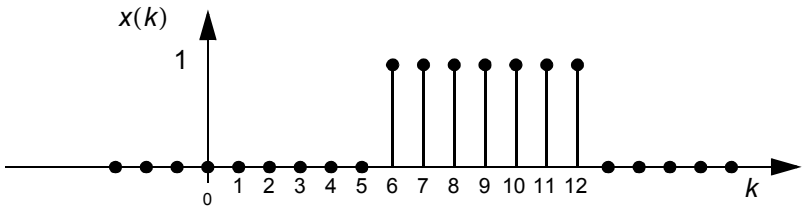
$$\text{rect}((k-k_0)/\kappa) = \begin{cases} 1 \text{ if } |k-k_0| < \kappa/2 \\ 0 \text{ otherwise} \end{cases} \quad \text{discrete time} \tag{496}$$



The *continuous rectangular pulse* $q(t) = \text{rect}((k-k_0)/\kappa)$

A *rectangular pulse* can also be generated by the addition of unit step functions. The unit step function is defined as:

$$u(k-k_0) = \begin{cases} 0 \text{ if } k < k_0 \\ 1 \text{ if } k \geq k_0 \end{cases} \quad \text{discrete time} \tag{497}$$



$x(k) = \text{rect}((k-9)/7) = u(k-4) - u(k-10)$

A *rectangular pulse train*, or square wave can be produced by distributing a rectangular pulse in a non-overlapping fashion. See also *Elementary Signals, Square Wave, Triangular Pulse, Unit Step Function*.

Rectangular Pulse Train: See *Square Wave*.

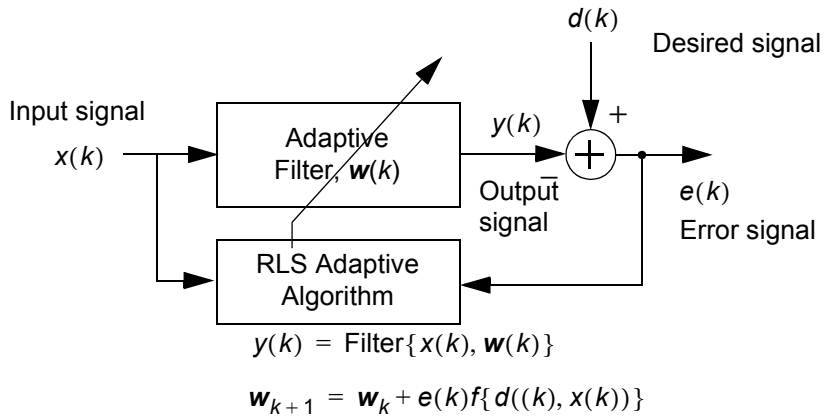
Recursive LMS: See *Least Mean Squares IIR Algorithms*.

Red Book: The specifications for the compact disc (CD) digital audio format were jointly specified by Sony and Philips and are documented in what is known as the *Red Book*. The standards for CD are also documented in the IEC (International Electrotechnical Commission) standard BNN15-83-095, and IEC-958 and IEC-908 .

Reed Solomon Coding: See *Cross Interleaved Reed Solomon Coding*.

Recruitment: See *Loudness Recruitment*.

Recursive Least Squares (RLS): The *RLS* algorithm can also be used to update the weights of an adaptive filter where the aim is to minimize the sum of the squared error signal. Consider the adaptive FIR digital filter which is to be updated using an RLS algorithm such that as new data arrives the *RLS* algorithm uses this new data (innovation) to improve the least squares solution:



For least squares adaptive signal processing the aim is to adapt the impulse response of the FIR digital filter such that the input signal $x(k)$ is filtered to produce $y(k)$ which when subtracted from desired signal $d(k)$, minimises the sum of the squared error signal $e(k)$ over time from the start of the signal at 0 (zero) to the current time k.

Note: While the above figure is reminiscent of the Least Mean Squares (LMS) adaptive filter, the distinction between the two approaches is quite important: LMS minimizes the *mean* of the square of the output error, while *RLS* minimizes the actual sum of the squared output errors.

In order to minimize the error signal, $e(k)$, consider minimizing the total sum of squared errors for all input signals up to and including time, k. The total squared error, $v(k)$, is:

$$v(k) = \sum_{s=0}^{k} [e(s)]^2 = e^2(0) + e^2(1) + e^2(2) + \ldots + e^2(k) \qquad (498)$$

Using vector notation, the error signal can be expressed in a vector format and therefore:

$$\boldsymbol{e}_k = \begin{bmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(k-1) \\ e(k) \end{bmatrix} = \begin{bmatrix} d(0) \\ d(1) \\ d(2) \\ \vdots \\ d(k-1) \\ d(k) \end{bmatrix} - \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(k-1) \\ y(k) \end{bmatrix} = \boldsymbol{d}_k - \boldsymbol{y}_k \tag{499}$$

Noting that the output of the N weight adaptive FIR digital filter is given by:

$$y(k) = \sum_{n=0}^{N-1} w_n x(k-n) = \boldsymbol{w}^T \boldsymbol{x}_k = \boldsymbol{x}_k^T \boldsymbol{w} \tag{500}$$

where,

$$\boldsymbol{w} = [w_0, w_1, w_2, \ldots, w_{N-1}] \text{ and} \tag{501}$$

$$\boldsymbol{x}_k = [x(k), x(k-1), x(k-2), \ldots, x(k-N+1)] \tag{502}$$

then Eq. 499 can be rearranged to give:

$$\begin{aligned} \boldsymbol{e}_k &= \begin{bmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(k-1) \\ e(k) \end{bmatrix} = \boldsymbol{d}_k - \begin{bmatrix} \boldsymbol{x}_0^T \boldsymbol{w} \\ \boldsymbol{x}_1^T \boldsymbol{w} \\ \boldsymbol{x}_2^T \boldsymbol{w} \\ \vdots \\ \boldsymbol{x}_{k-1}^T \boldsymbol{w} \\ \boldsymbol{x}_k^T \boldsymbol{w} \end{bmatrix} = \boldsymbol{d}_k - \begin{bmatrix} \boldsymbol{x}_0^T \\ \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_{k-1}^T \\ \boldsymbol{x}_k^T \end{bmatrix} \boldsymbol{w} \\ &= \boldsymbol{d}_k - \begin{bmatrix} x(0) & 0 & 0 & \ldots & 0 \\ x(1) & x(0) & 0 & \ldots & 0 \\ x(2) & x(1) & x(0) & \ldots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ x(k-1) & x(k-2) & x(k-3) & \ldots & x(k-N) \\ x(k) & x(k-1) & x(k-2) & \ldots & x(k-N+1) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{N-1} \end{bmatrix} \end{aligned} \tag{503}$$

i.e. $\boldsymbol{e}_k = \boldsymbol{d}_k - \boldsymbol{X}_k \boldsymbol{w}$

where $\boldsymbol{X}_k$ is a $(k+1) \times N$ data matrix made up from input signal samples. Note that the first N rows of $\boldsymbol{X}_k$ are sparse. Equation 498 can be rewritten such that:

$$\begin{aligned} v(k) &= \boldsymbol{e}_k^T \boldsymbol{e}_k = \|\boldsymbol{e}_k\|_2^2 \\ &= [\boldsymbol{d}_k - \boldsymbol{X}_k \boldsymbol{w}]^T [\boldsymbol{d}_k - \boldsymbol{X}_k \boldsymbol{w}] \\ &= \boldsymbol{d}_k^T \boldsymbol{d}_k + \boldsymbol{w}^T \boldsymbol{X}_k^T \boldsymbol{X}_k \boldsymbol{w} - 2\boldsymbol{d}_k^T \boldsymbol{X}_k \boldsymbol{w} \end{aligned} \tag{504}$$

where $\|\boldsymbol{e}_k\|_2$ is the 2-norm of the vector $\boldsymbol{e}_k$. From a first glance at the last line of Eq. 503 it may seem that a viable solution is to set $\boldsymbol{e}_k = \boldsymbol{0}$ then simply solve the equation $\boldsymbol{w} = \boldsymbol{X}_k^{-1} \boldsymbol{d}_k$. However this is of course not possible in general as $\boldsymbol{X}_k$ is not a square matrix and therefore ***not*** invertible.

In order to find a "good" solution such that the 2-norm of the error vector, $\boldsymbol{e}_k$, is minimized, note that Eq. 504 is quadratic in the vector $\boldsymbol{w}$, and the function $v(k)$ is an up-facing *hyperparaboloid* when plotted in N+1 dimensional space, and there exists exactly one minimum point at the bottom of the hyperparaboloid where the gradient vector is zero, i.e.,

$$\frac{\partial}{\partial \boldsymbol{w}} v(k) = \boldsymbol{0} \tag{505}$$

From Eq. 504

$$\frac{\partial}{\partial \boldsymbol{w}} v(k) = 2\boldsymbol{X}_k^T \boldsymbol{X}_k \boldsymbol{w} - 2\boldsymbol{X}_k^T \boldsymbol{d}_k = -2\boldsymbol{X}_k^T [\boldsymbol{d}_k - \boldsymbol{X}_k \boldsymbol{w}] \tag{506}$$

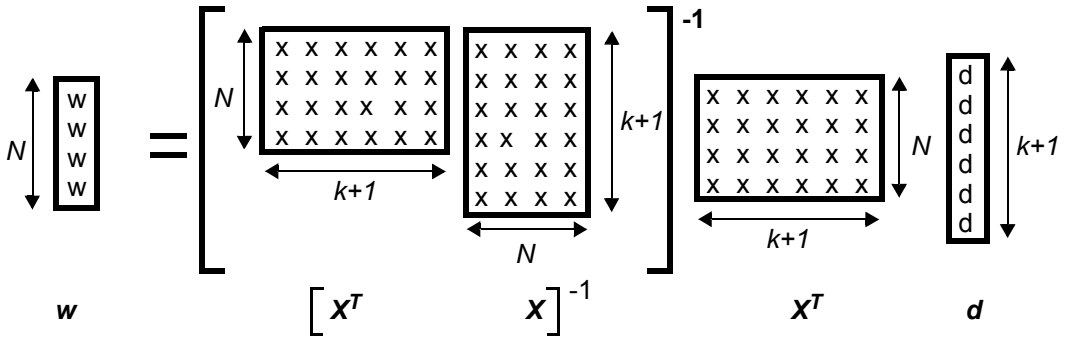
and therefore:

$$\begin{aligned} -2\boldsymbol{X}_k^T [\boldsymbol{d}_k - \boldsymbol{X}_k \boldsymbol{w}_{LS}] &= \boldsymbol{0} \\ \Rightarrow \boldsymbol{X}_k^T \boldsymbol{X}_k \boldsymbol{w}_{LS} &= \boldsymbol{X}_k^T \boldsymbol{d}_k \end{aligned} \tag{507}$$

and the least squares solution, denoted as $\boldsymbol{w}_{LS}$ and based on data received up to and including time, k, is given as:

$$\boldsymbol{w}_{LS} = [\boldsymbol{X}_k^T \boldsymbol{X}_k]^{-1} \boldsymbol{X}_k^T \boldsymbol{d}_k \tag{508}$$

Note that because $[\boldsymbol{X}_k^T \boldsymbol{X}_k]$ is a symmetric square matrix, then $[\boldsymbol{X}_k^T \boldsymbol{X}_k]^{-1}$ is also a symmetric square matrix. As with any linear algebraic manipulation a useful check is to confirm that the matrix dimensions are compatible, thus ensuring that $\boldsymbol{w}_{LS}$ is a $N \times 1$ matrix:

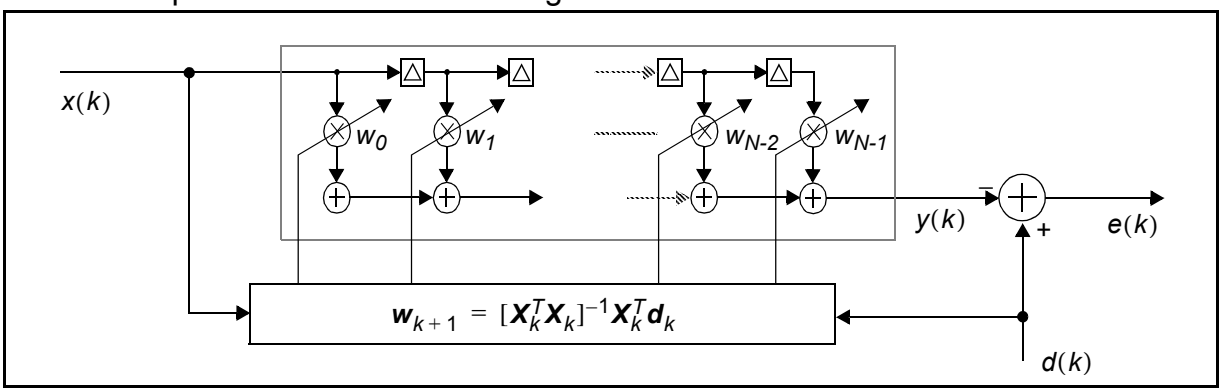


Note that if in the special case where $\boldsymbol{X}_k$ is a square non-singular matrix, then Eq. 508 simplifies to:

$$\boldsymbol{w}_{LS} = \boldsymbol{X}_k^{-1}\boldsymbol{X}_k^{-T}\boldsymbol{X}_k^T\boldsymbol{d}_k = \boldsymbol{X}_k^{-1}\boldsymbol{d}_k \tag{509}$$

The computation to calculate Eq. 508 requires about $O(N^4)$ MACs (multiply/accumulates) and $O(N)$ divides for the matrix inversion, and $O((k+1)\times N^2)$ MACs for the matrix multiplications. Clearly therefore, the more data that is available, then the more computation required.

At time iteration k+1, the weight vector to use in the adaptive FIR filter that minimizes the 2-norm of the error vector, $\boldsymbol{e}_k$ can be denoted as $\boldsymbol{w}_{k+1}$, and the open loop least squares adaptive filter solution can be represented as the block diagram:



Note however that at time $k+1$ when a new data sample arrives at both the input, $x(k+1)$, and the desired input, $d(k+1)$ then this new information should ideally be incorporated in the least squares solution with a view to obtaining an improved solution. The new least squares filter weight vector to use at time $k+2$ (denoted as $\boldsymbol{w}_{k+2}$) is clearly given by:

$$\boldsymbol{w}_{k+2} = [\boldsymbol{X}_{k+1}^T\boldsymbol{X}_{k+1}]^{-1}\boldsymbol{X}_{k+1}^T\boldsymbol{d}_{k+1} \tag{510}$$

This equation requires that another full matrix inversion is performed, $[\boldsymbol{X}_{k+1}^T\boldsymbol{X}_{k+1}]^{-1}$, followed by the appropriate matrix multiplications. This very high level of computation for every new data sample provides the motivation for deriving the *recursive least squares* (RLS) algorithm. *RLS* has a much lower level of computation by calculating $\boldsymbol{w}_{k+1}$ using the result of previous estimate $\boldsymbol{w}_k$ to reduce computation.

Consider the situation where we have calculated $\boldsymbol{w}_k$, from,

$$\boldsymbol{w}_k = [\,\boldsymbol{X}_{k-1}^T\boldsymbol{X}_{k-1}]^{-1}\boldsymbol{X}_{k-1}^T\boldsymbol{d}_{k-1} = \boldsymbol{P}_{k-1}\boldsymbol{X}_{k-1}^T\boldsymbol{d}_{k-1} \tag{511}$$

where

$$\boldsymbol{P}_{k-1} = [\,\boldsymbol{X}_{k-1}^T\boldsymbol{X}_{k-1}]^{-1} \tag{512}$$

When the new data samples, $x(k)$ and $d(k)$, arrive we have to calculate:

$$\boldsymbol{w}_{k+1} = [\,\boldsymbol{X}_k^T\boldsymbol{X}_k]^{-1}\boldsymbol{X}_k^T\boldsymbol{d}_k = \boldsymbol{P}_k\boldsymbol{X}_k^T\boldsymbol{d}_k \tag{513}$$

However note that $\boldsymbol{P}_k$ can be written in terms of the previous data matrix $\boldsymbol{X}_{k-1}$ and the data vector $\boldsymbol{x}_k$ by partitioning the matrix $\boldsymbol{X}_k$:

$$\begin{aligned} \boldsymbol{P}_k &= [\,\boldsymbol{X}_k^T\boldsymbol{X}_k]^{-1} = \begin{bmatrix}\boldsymbol{X}_{k-1}^T & \boldsymbol{x}_k\end{bmatrix}\begin{bmatrix}\boldsymbol{X}_{k-1} \\ \boldsymbol{x}_k^T\end{bmatrix} \\ &= \left[\boldsymbol{X}_{k-1}^T\boldsymbol{X}_{k-1} + \boldsymbol{x}_k\boldsymbol{x}_k^T\right]^{-1} \\ &= \left[\boldsymbol{P}_{k-1}^{-1} + \boldsymbol{x}_k\boldsymbol{x}_k^T\right]^{-1} \end{aligned} \tag{514}$$

where, of course, $\boldsymbol{x}_k = [x(k+1), x(k), x(k-1), \ldots, x(k-N+1)]$ as before in Eq. 500. In order to write Eq. 514 in a more "suitable form" we use the matrix inversion lemma (see *Matrix Properties-Inversion Lemma*) which states that:

$$[\boldsymbol{A}^{-1} + \boldsymbol{BCD}]^{-1} = \boldsymbol{A} - \boldsymbol{AB}[\boldsymbol{C} + \boldsymbol{DAB}]^{-1}\boldsymbol{DA} \tag{515}$$

where $\boldsymbol{A}$ is a non-singular matrix and $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are appropriately dimensioned matrices. Using the matrix inversion lemma of Eq.514, where $\boldsymbol{P}_{k-1} = \boldsymbol{A}$, $\boldsymbol{x}_k = \boldsymbol{B}$, $\boldsymbol{x}_k^T = \boldsymbol{D}$ and $\boldsymbol{C}$ is the 1×1 identity matrix. i.e. the scalar 1, then:

$$\boldsymbol{P}_k = \boldsymbol{P}_{k-1} - \boldsymbol{P}_{k-1}\boldsymbol{x}_k[1 + \boldsymbol{x}_k^T\boldsymbol{P}_{k-1}\boldsymbol{x}_k]^{-1}\boldsymbol{x}_k^T\boldsymbol{P}_{k-1} \tag{516}$$

This equation implies that if we know the matrix $[\boldsymbol{X}_{k-1}^T\boldsymbol{X}_{k-1}]^{-1}$ then the matrix $[\boldsymbol{X}_k^T\boldsymbol{X}_k]^{-1}$ can be computed without explicitly performing a complete matrix inversion from first principles. This, of course, saves in computation effort. Equations 513 and 516 are one form of the RLS algorithm. By additional algebraic manipulation, the computation complexity of Eq. 516 can be simplified even further.

By substituting Eq. 516 into Eq. 513, and partitioning the vector $\boldsymbol{d}_k$ and simplifying gives:

$$\begin{aligned} \boldsymbol{w}_{k+1} &= [\boldsymbol{P}_{k-1} - \boldsymbol{P}_{k-1}\boldsymbol{x}_k[1 + \boldsymbol{x}_k^T\boldsymbol{P}_{k-1}\boldsymbol{x}_k]^{-1}\boldsymbol{x}_k^T\boldsymbol{P}_{k-1}]\boldsymbol{X}_k^T\boldsymbol{d}_k \\ &= [\boldsymbol{P}_{k-1} - \boldsymbol{P}_{k-1}\boldsymbol{x}_k[1 + \boldsymbol{x}_k^T\boldsymbol{P}_{k-1}\boldsymbol{x}_k]^{-1}\boldsymbol{x}_k^T\boldsymbol{P}_{k-1}]\begin{bmatrix}\boldsymbol{X}_{k-1}^T & \boldsymbol{x}_k\end{bmatrix}\begin{bmatrix}\boldsymbol{d}_{k-1} \\ d(k)\end{bmatrix} \\ &= [\boldsymbol{P}_{k-1} - \boldsymbol{P}_{k-1}\boldsymbol{x}_k[1 + \boldsymbol{x}_k^T\boldsymbol{P}_{k-1}\boldsymbol{x}_k]^{-1}\boldsymbol{x}_k^T\boldsymbol{P}_{k-1}]\left[\boldsymbol{X}_{k-1}^T\boldsymbol{d}_{k-1} + \boldsymbol{x}_k d(k)\right] \end{aligned} \tag{517}$$

Using the substitution that $\boldsymbol{w}_k = \boldsymbol{P}_{k-1}\boldsymbol{X}_{k-1}^T\boldsymbol{d}_{k-1}$ and also dropping the time subscripts for notational convenience, i.e. $\boldsymbol{P} = \boldsymbol{P}_{k-1}$, $\boldsymbol{x} = \boldsymbol{x}_k$, $\boldsymbol{d} = \boldsymbol{d}_{k-1}$, and $d = d(k)$, further simplification can be performed:

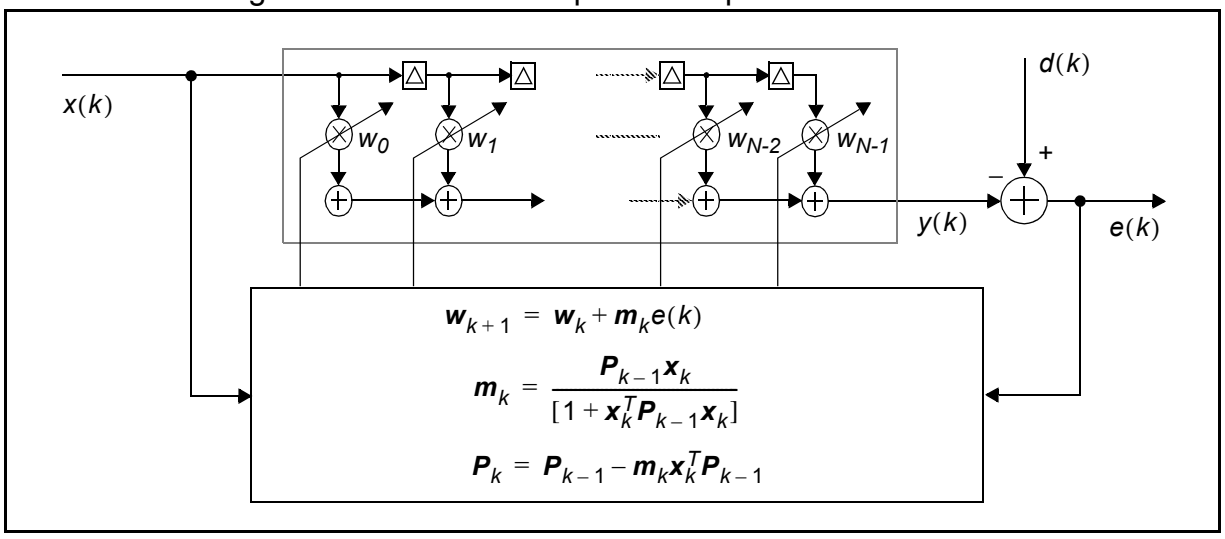
$$
\begin{aligned}
\boldsymbol{w}_{k+1} &= [\boldsymbol{P} - \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}\boldsymbol{x}^T\boldsymbol{P}]\left[\boldsymbol{X}^T\boldsymbol{d} + \boldsymbol{x}d\right] \\
&= \boldsymbol{P}\boldsymbol{X}^T\boldsymbol{d} + \boldsymbol{P}\boldsymbol{x}d - \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{X}^T\boldsymbol{d} - \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}d \\
&= \boldsymbol{w}_k - \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}\boldsymbol{x}^T\boldsymbol{w}_k + \boldsymbol{P}\boldsymbol{x}d - \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}d \\
&= \boldsymbol{w}_k - \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}\boldsymbol{x}^T\boldsymbol{w}_k + \boldsymbol{P}\boldsymbol{x}d[1 - [1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}] \\
&= \boldsymbol{w}_k - \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}\boldsymbol{x}^T\boldsymbol{w}_k + \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}[[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}] - \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]d \\
&= \boldsymbol{w}_k - \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}\boldsymbol{x}^T\boldsymbol{w}_k + \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}d \\
&= \boldsymbol{w}_k + \boldsymbol{P}\boldsymbol{x}[1 + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}]^{-1}(d - \boldsymbol{x}^T\boldsymbol{w}_k)
\end{aligned}
\tag{518}
$$

and reintroducing the subscripts, and noting that $y(k) = \boldsymbol{x}_k^T\boldsymbol{w}_k$:

$$
\begin{aligned}
\boldsymbol{w}_{k+1} &= \boldsymbol{w}_k + \boldsymbol{P}_{k-1}\boldsymbol{x}_k[1 + \boldsymbol{x}_k^T\boldsymbol{P}_{k-1}\boldsymbol{x}_k]^{-1}(d(k) - y(k)) \\
&= \boldsymbol{w}_k + \boldsymbol{m}_k(d(k) - y(k)) \\
&= \boldsymbol{w}_k + \boldsymbol{m}_k e(k)
\end{aligned}
\tag{519}
$$

where $\boldsymbol{m}_k = \boldsymbol{P}_{k-1}\boldsymbol{x}_k[1 + \boldsymbol{x}_k^T\boldsymbol{P}_{k-1}\boldsymbol{x}_k]^{-1}$ and is called the gain vector.

The *RLS* adaptive filtering algorithm therefore requires that at each time step, the vector $\boldsymbol{m}_k$ and the matrix $\boldsymbol{P}_k$ are computed. The filter weights are then updated using the error output, $e(k)$. Therefore the block diagram for the closed loop *RLS* adaptive FIR filter is: :



The above form of the *RLS* requires $O(N^2)$ MACs and one divide on each iteration. See also *Adaptive Filtering, Least Mean Squares Algorithm, Least Squares, Noise Cancellation, Recursive Least Squares-Exponentially Weighted.*

Recursive Least Squares (RLS) - Exponentially Weighted: One problem with least squares and *recursive least squares (RLS)* algorithm derived in entry *Recursive Least Squares*, is that the minimization of the 2-norm of the error vector $\boldsymbol{e}_k$ calculates the least squares vector at time *k* based on all previous data, i.e. data from long ago is given as much relevance as recently received data. Therefore if at some time in the past a block of "bad" data was received or the input signal statistics changed then the *RLS* algorithm will calculate the current least squares solution giving as much

relevance to the old (and probably irrelevant) data as it does to very recent inputs. Therefore the *RLS* algorithm has infinite memory.

In order to overcome the infinite memory problem, the exponentially weighted least squares, and exponentially weighted recursive least squares (EW-RLS) algorithms can be derived. Consider again Eq. 498 where this time each error sample is weighted using a forgetting factor constant λ which just less than 1:

$$v(k) = \sum_{s=0}^{k} \lambda^{k-s}[e(s)]^2 = \lambda^k e^2(0) + \lambda^{k-1} e^2(1) + \lambda^{2k-2} e^2(2) + \ldots + e^2(k) \tag{520}$$

For example if a forgetting factor of 0.9 was chosen then data which is 100 time iterations old is pre-multiplied by $0.9^{100} = 2.6561 \times 10^{-5}$ and thus considerably de-emphasized compared to the current data. Therefore in dB terms, data that is more 100 time iterations old is attenuated by $10\log(0.00026561) = -46$ dB. Data that is more than 200 time iterations old is therefore attenuated by around 92 dB, and if the input data were 16 bit fixed point corresponding to a dynamic range of 96dB, then the old data is on the verge of being completely forgotten about. The forgetting factor is typically a value of between 0.9 and 0.9999.

Noting the form of Eq. 504 we can rewrite Eq. 520 as:

$$v(k) = \boldsymbol{e}_k^T \Lambda_k \boldsymbol{e}_k \tag{521}$$

where Λ_k is a $(k+1)\times(k+1)$ diagonal matrix $\Lambda_k = \text{diag}[\lambda^k, \lambda^{k-1}, \lambda^{k-2}, \ldots, \lambda, 1]$

Therefore:

$$\begin{aligned} v(k) &= [\boldsymbol{d}_k - \boldsymbol{X}_k \boldsymbol{w}]^T \Lambda_k [\boldsymbol{d}_k - \boldsymbol{X}_k \boldsymbol{w}] \\ &= \boldsymbol{d}_k^T \Lambda_k \boldsymbol{d}_k + \boldsymbol{w}^T \boldsymbol{X}_k^T \Lambda_k \boldsymbol{X}_k \boldsymbol{w} - 2\boldsymbol{d}_k^T \Lambda_k \boldsymbol{X}_k \boldsymbol{w} \end{aligned} \tag{522}$$

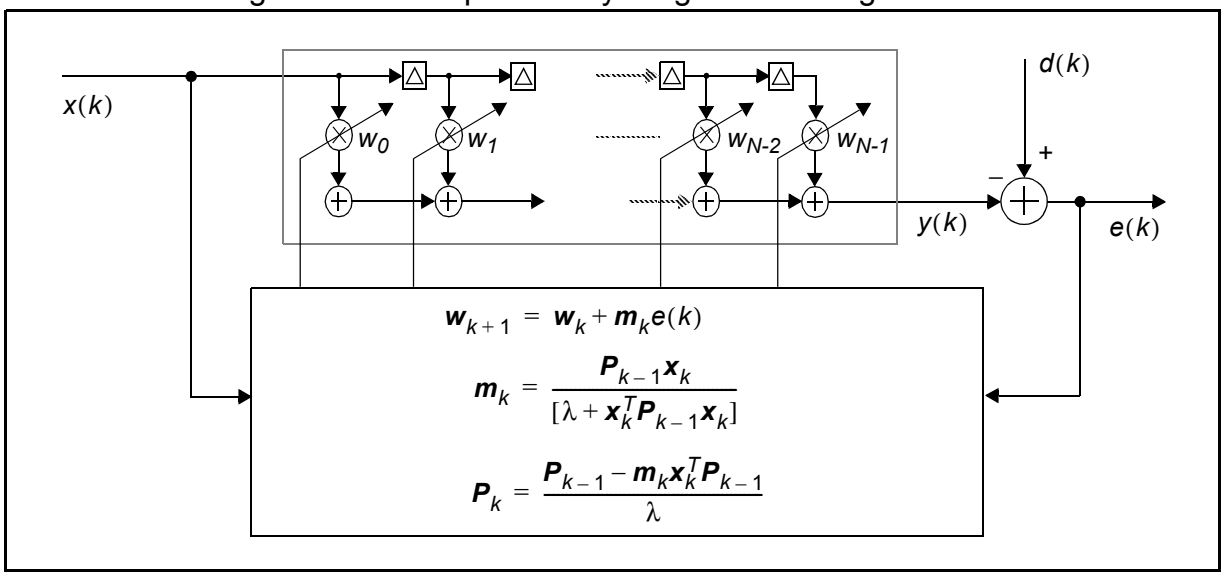
Following the same procedure as for Eqs. 505 to 508 the exponentially weight least squares solution is easily found to be:

$$\boldsymbol{w}_{LS} = [\boldsymbol{X}_k^T \Lambda_k \boldsymbol{X}_k]^{-1} \boldsymbol{X}_k^T \Lambda_k \boldsymbol{d}_k \tag{523}$$

In the same way as the RLS algorithm was realised, we can follow the same approach as Eqs. 511 to 519 and realise the exponentially weighted RLS algorithm:

$$\begin{aligned} \boldsymbol{w}_{k+1} &= \boldsymbol{w}_k + \boldsymbol{m}_k e(k) \\ \boldsymbol{m}_k &= \frac{\boldsymbol{P}_{k-1}\boldsymbol{x}_k}{[\lambda + \boldsymbol{x}_k^T \boldsymbol{P}_{k-1} \boldsymbol{x}_k]} \\ \boldsymbol{P}_k &= \frac{\boldsymbol{P}_{k-1} - \boldsymbol{m}_k \boldsymbol{x}_k^T \boldsymbol{P}_{k-1}}{\lambda} \end{aligned} \tag{524}$$

Therefore the block diagram for the exponentially weighted RLS algorithm is:



Compared to the Least Mean Squares (LMS) algorithm, the RLS can provide much faster convergence and a smaller error, however the computation required is a factor of *N* more than for the LMS, where *N* is the adaptive filter length. The RLS is less numerically robust than the LMS. For more detailed information refer to [77]. See also *Adaptive Filtering, Least Mean Squares Algorithm, Least Squares, Noise Cancellation, Recursive Least Squares.*

Reflection: Sound can be *reflected* when a sound wave reaches a propagation medium boundary, e.g. from air to brick (wall). Some of the sound may be reflected and the rest will either be absorbed (converted to heat or transmitted through the medium). See also *Absorption*.

Register: A memory location inside a DSP processor, used for temporary storage of data. Access to the data in a *register* is very fast as no off-chip memory movements are required.

Relative Error: The ratio of the absolute error (difference between true value and estimated value) to the true value of a particular quantity is called the *relative error*. For example consider two real numbers *x* and *y*, that will be represented to only one decimal place of precision:

$$x = 1.345 \text{ and} \tag{525}$$

$$y = 1000.345 \tag{526}$$

The rounded values, denoted as *x'* and *y'* will be given by

$$x' = 1.3 \text{ and} \tag{527}$$

$$y' = 1000.3 \tag{528}$$

The absolute errors, Δx and Δy, caused by the rounding are the same for both quantities, and given by:

$$\Delta x = x - x' = 1.345 - 1.3 = 0.045 \tag{529}$$

$$\Delta y = y - y' = 1000.345 - 1000.3 = 0.045 \tag{530}$$

The relative error, however, is defined as the ratio of the absolute error to the correct value. Therefore the *relative error* of *x'* and *y'* can be calculated as:

$$\frac{\Delta x}{x} = \frac{0.045}{1.345} = 0.0334 \tag{531}$$

$$\frac{\Delta y}{y} = \frac{0.045}{1000.345} = 4.5 \times 10^{-5} \tag{532}$$

Relative error is often denoted as a **percentage error**. Therefore in the above example *x'* represents a 3.34% error, whereas *y'* is only a 0.0045% error. *Relative errors* are widely used in error analysis calculations where the results of computations on estimated, rounded or truncated quantities can be predicted by manipulating only the *relative errors*. See also *Absolute Error, Error Analysis*.

Relative Pitch: The ability to specify the names of musical notes on the Western music scale if the name of one of the notes is first given is known as *relative pitch*. *Relative pitch* skills are relatively common among singers and musicians. The ability to identify any musical note with no clues is known as perfect or absolute pitch and is less common. See also *Music, Perfect Pitch, Pitch, Western Music Scale*.

Resistor-Capacitor Circuit: See *RC Circuit*.

Resolution: The accuracy to which a particular quantity has been converted. If the resolution of a particular A/D converter is 10mVolts then this means that every analog quantity is *resolved* to within 10mVolts of its true value after conversion.

Resonance: When an object is vibrating at its resonant frequency it is said to be in *resonance*. See *Resonant Frequency.*

Resonant Frequency: All mechanical objects have a *resonant* or natural frequency at which they will vibrate if excited by an impulse. For example, striking a bell, or other metal object will, cause a ringing sound (derived from the vibrations) at the bell's resonant or natural frequency. If a component is excited by vibrations at its *resonant frequency* then it will start to vibrate in synchrony and lead to vibrations of a very large magnitude. This is referred to as sympathetic vibration. For example, if a tone at the same frequency as a bell's *resonant frequency* is played nearby, the bell will start to ring in unison at the same frequency. Music is derived from instruments' vibrating strings and membranes, and columns of air at *resonant frequency*.

Resource Interchange File Format (RIFF): *RIFF* is a proprietary format developed by IBM and Microsoft. *RIFF* essentially defines a set of file formats which are suitable for multimedia file handling (i.e. audio, video, and graphics):

- Playing back multimedia data;
- Recording multimedia data;
- Exchanging multimedia data between applications and across platforms.

A *RIFF* file is composed of a descriptive header identifying the type of data, the size of the data, and the actual data. Currently well known forms of *RIFF* file are:

- WAVE: Waveform Audio Format (.WAV files)

- PAL: Palette File Format (.PAL files)
- RDIB: RIFF Device Independent Bitmap Format (.DIB files)
- RMID: RIFF MIDI Format (.MID files)
- RMMP: RIFF Multimedia Movie File Format

RIFF files are supported by Microsoft Windows on the PC. (Note that there is also a counterpart to *RIFF* called RIFX that uses the Motorola integer byte ordering format rather than the Intel format.) See also *Standards*.

Return to Zero: See *Non-Return to Zero*.

Reverberation: The multitude of a particular sound's waves that add to the direct path sound wave but slightly later in time due to the longer distance (reflected) transmission paths. Virtually all rooms have some level of *reverberation* (compare a carpeted office to an indoor swimming pool to contrast rooms with short *reverberation* time to those with long *reverberation* times.) More formally the reverberation time in a room is defined as the time it takes a sound to fall to one millionth (reduce by 60dB) of its initial sound intensity.

Ringing Tone: Tones at 440 Hz and 480 Hz make up the *ringing tone* for telephone systems. See also *DialTone, Dual Tone Multifrequency*.

Ripple Adder: See *Parallel Adder*.

RISC: *RISC* (Reduced instruction set computer) refers to a microprocessor that has implemented a smaller core of instructions than a Complex Instruction Set Computer (CISC) in order that the silicon area can be filled with more application appropriate facilities. Some designers refer to DSP processors are *RISC*, whereas others note that *RISCs* are subtly different and lack features such as internal DMA, multiple interrupt pins, single cycle MACs, wide accumulators and so on. *RISCs* are designed to perform a wide range of general purpose instructions unlike DSPs, which are optimized for MACs. Texas Instruments describe their TMS320C31 DSP chip as a hybrid DSP, with features of both *RISC* and CISC. Best not to worry!

RS232: A simple serial communications protocol. A few DSP boards use *RS232* lines to communicate with the host computer. The ITU (formerly CCITT) adopted a related version of the *RS232* cable which is specified in recommendation V24.

Robinson-Dadson Curves: *Robinson* and *Dadson's* 1956 paper [126] studied the definition of sound intensity, the subjective loudness of human hearing, and associated audiometric measurements. They repeated elements of earlier work by Fletcher and Munson in 1933 [73] and produced a set of equal loudness contours which showed the variation in sound pressure level (SPL) of tones at different frequencies that are perceived as having the same loudness. See also *Equal Loudness Contours, Frequency Range of Hearing, Loudness Recruitment, Sound Pressure Level, Threshold of Hearing*.

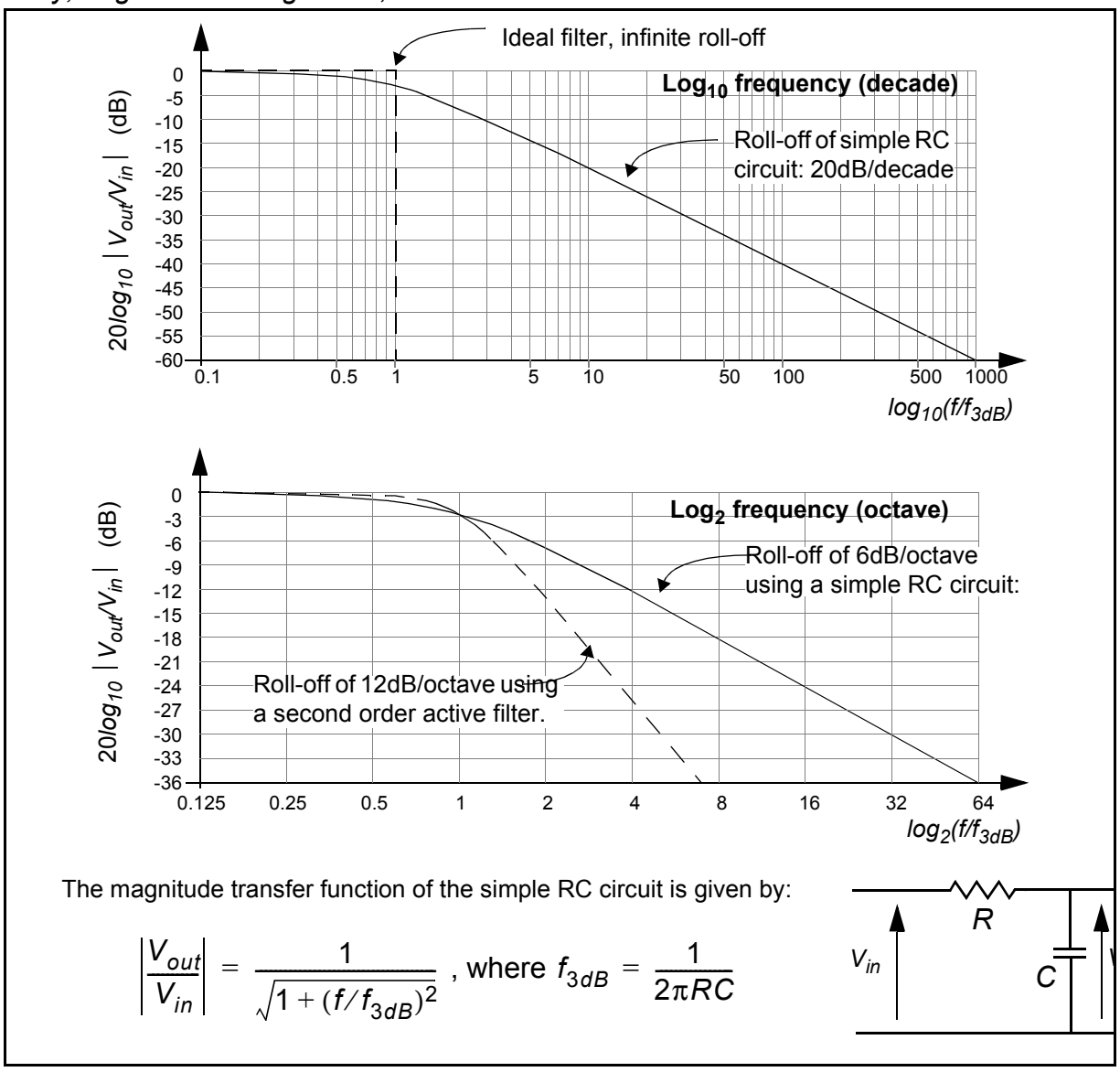
Roll-off: Common filter types such as low pass, band pass, or high pass filters have distinct regions: the passband, transition band(s) and stopband(s). The increasing attenuation above the 3dB point from the passband to the stopband is referred to as the transition band. The rate at which the filter response decreases from passband to stop band is called the *roll-off* of the filter. The higher the *roll-off*, then the closer the filter is to the ideal filter which would have an infinite *roll-off* from passband to stopband.

The *roll-off* a simple analog (single pole) RC circuit is 6dB/octave at frequencies above the cut-off frequency, f_{3dB}, (or 3dB point). If two RC circuits are cascaded together to realise a second order (two pole) filter then the *roll-off* at frequencies above the cut-off frequency will be 12dB/octave or 40dB/decade (To attain better *roll-off* is it unlikely that passive RC circuits would be cascaded together, and it is more likely that a higher order active filter would be used). In general for an *N*-th order/pole cascaded RC filter (and which will have at least *N* capacitors), the roll-off rate at frequencies high above f_{3dB} the *roll-off* will be:

$$\begin{aligned}\text{Roll-off} &= 20\log_{10}\left(\frac{1}{\sqrt{1+a_o(f/f_{3dB})^2+\ldots+(f/f_{3dB})^{2N}}}\right) \\ &\approx -20N\log_{10}(f/f_{3dB})\end{aligned} \tag{533}$$

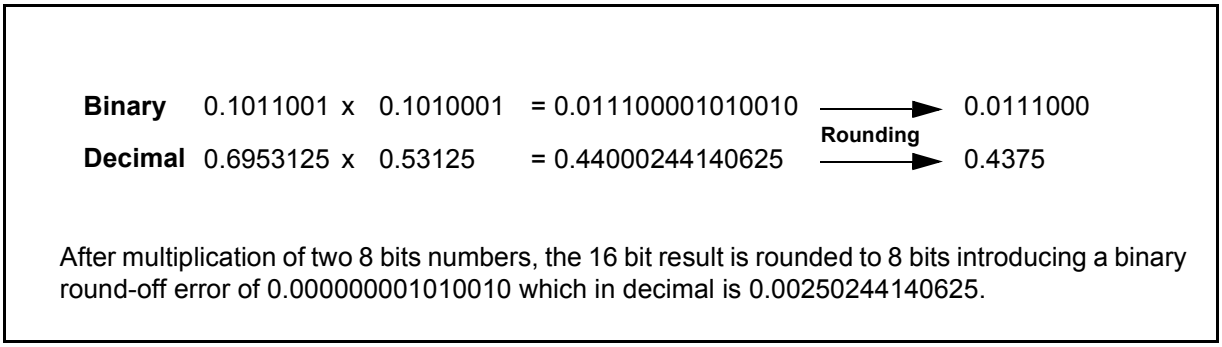
For applications such as analog anti-alias filters, Bessel, Butterworth or Chebychev filters with sharp cut-off frequencies with a hard knee at f_{3dB} are required and the *roll-off* rate should be at least the same as the dynamic range of the digital wordlength. For example using an ADC with 16 bits wordlength and dynamic range $20\log 2^{16} = 96\text{dB}$ it would be advisable to use an anti-alias filter of *at least* 96dB/octave such that any frequency components above f_s are completed removed. Note that even with this sharp cut-off some frequency components between $f_s/2$ and f_s will still alias down to the baseband if f_{3dB} is chosen to equal $f_s/2$. If less selective filters are available, it is generally necessary to set f_{3dB} to less than $f_s/2$ (or use oversampling techniques). See also *Active Filter,*

Decade, Decibels, Filter (Bessel, Butterworth, Chebychev), Knee (of a filter), Logarithmic Frequency, Logarithmic Magnitude, *Octave*. .



Round-Off Error: When two *N* bit numbers are multiplied together, the result is a number with 2*N* bits. If a fixed point DSP processor with *N* bits resolution is used, the 2*N* bit number cannot be accommodated for future computations which can operate on only *N* bit operands. Therefore, if we assume that the original *N* bit numbers were both constrained to be less than 1 in magnitude by using a binary point, then the 2*N* bit result is also less that 1. Hence if we round the least significant *N* bits up or down, then this is equivalent to losing precision. This loss of precision is referred to as *round-off error*. Although the *round-off error* for a single computation is usually not significant, many errors added together can be significant. Furthermore if the result of a computation yields the value

of 0 (zero) after *rounding*, and this result is to be used as a divisor, a divide by zero error will occur. See also *Truncation Error, Fractional Binary, Binary Point*.


Binary 0.1011001 x 0.1010001 = 0.011100001010010
0.0111000
Rounding
Decimal 0.6953125 x 0.53125 = 0.44000244140625
0.4375
After multiplication of two 8 bits numbers, the 16 bit result is rounded to 8 bits introducing a binary round-off error of 0.000000001010010 which in decimal is 0.00250244140625.

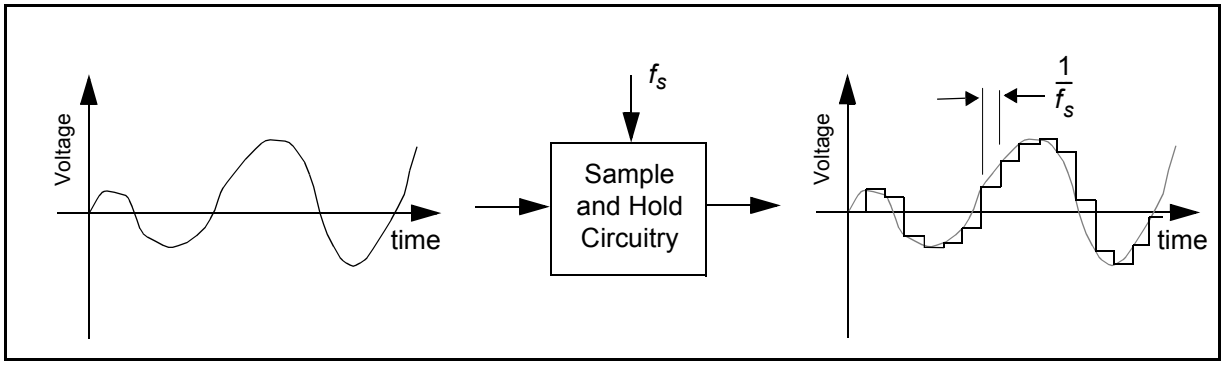
Round-Off Noise: When round-off errors are modelled as a source of additive noise in a system, the effect is referred to as *round-off noise*. This noise is usually discussed in terms of its mean power. See also *Round-off Error.*

Row Vector: See *Vector*.

Run Length Encoding (RLE): If a data sequence contains a consecutive sequence of the same data word, then this is referred to as a "run", and the number of data words is referred to as the "length" of the run. *Run length encoding* is a technique that allows data sequences prone to repetitive values to be efficiently encoded and therefore compressed. For example, if a 256×256 image is stored in a file sequentially by each row, then a run of identical pixel values in a row can is encoded by two data words, one stating the repeated value, and one stating the length of the run. *Run length encoding* is a lossless compression technique. See also *Compression*.

S

Sample and Hold (S/H): A analog circuit used at the input to A/D converters to maintain a constant input voltage while the digital equivalent is calculated by the A/D converter. The output waveform is an analog voltage, that is "steppy" in appearance, with the duration of the steps (the hold time) being determined by the chosen sampling frequency f_s. The *sample and hold* function is also referred to as a zero order hold. See also *First Order Hold, Analog to Digital Converter*.

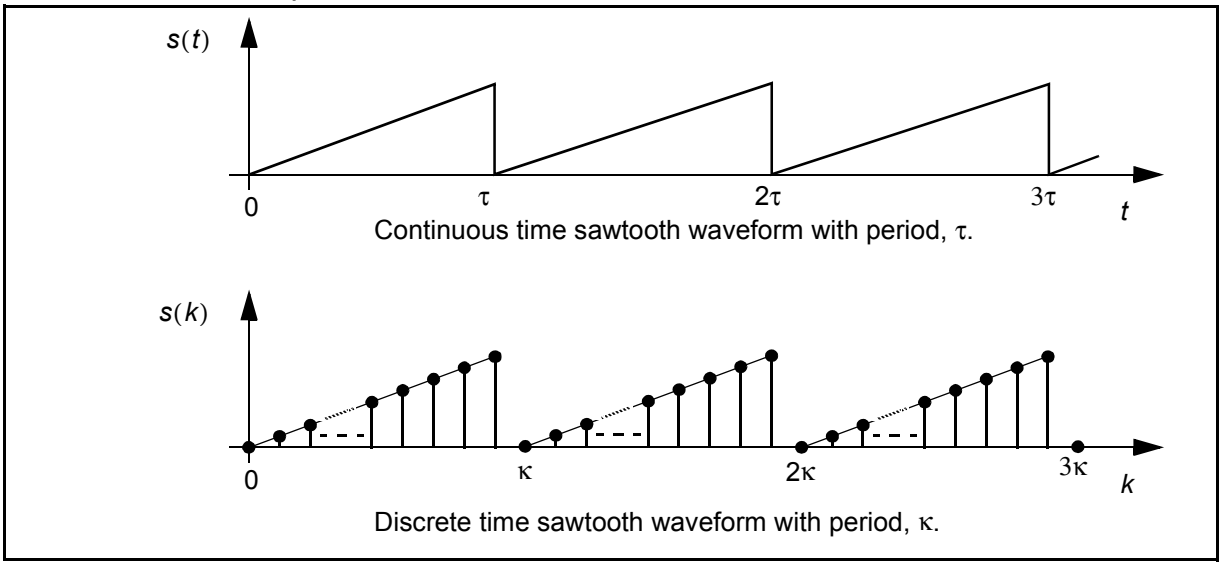


Sampling: The process of converting an analog signal into discrete samples at regular intervals. To correctly sample a signal the sampling rate or sampling frequency, f_s, should be at least twice the maximum frequency component of the signal (the Nyquist criteria). *Sampling* results in analog samples of a signal. Quantization converts these analog samples to a discrete set of values. See also *Analog to Digital Converter*.

Sampling Rate: The number of samples per second from a particular analog signal, usually expressed in Hz (Hertz).

Saturation Arithmetic: When the magnitude of the result of a computation will overflow the result is limited by the DSP processor to the maximum positive or negative number (otherwise the number would be too large for the processor wordlength). For a fixed point 16 bit DSP processor therefore, the maximum value generated by any computation will be 32767, and the minimum value will be -32768.

Sawtooth Waveform: A *sawtooth waveform* is a periodic signal made up from individual ramp waveforms. See also *Ramp Waveform*.



SAXPY: This term is used in vector algebra to indicate the calculation:

$$x = \alpha x + y \tag{534}$$

SAXPY is a mnemonic for *s*calar *a*lpha *x* *p*lus *y*, and has its origins as part of the Linpack software.[15]

Schur-Cohn Test: Given a z-domain polynomial of order *N*, the *Schur-Cohn test* can be used to establish if the roots of the polynomial are within the unit circle [77]. The *Schur-Cohn test* can therefore be used on IIR filters to check stability (i.e. all poles within the unit circle), or to test if a filter is minimum phase (all zeroes and poles within the unit circle).

Schur Form: See *Matrix Decompositions - Schur Form*.

Scrambler/Descrambler: A *scrambler* is either an analog or digital device used to implement secure communication channels by modifying a data stream or analog signal to appear random. A *descrambler* reverses the effect of the *scrambler* to recover the original signal. Many different techniques exist for scrambling signals and are of two main forms: frequency domain techniques, and time domain techniques.

Second Order: Usually meaning two of a particular device cascaded together. Used in a non-consistent way. *Second order* is often used to refer to a segment of a linear system that can be represented by a system polynomial of order 2.

Semitone: In music theory each adjacent note in the chromatic scale differs by one *semitone*, which corresponds to multiplying the lower frequency by the twelfth root of 2, i.e. $2^{1/12} = 1.0594631\ldots$. A difference of two *semitones* is a tone. See also *Western Music Scale*.

Semi-vowels: One of the elementary sounds of speech, namely plosives, fricatives, sibilant fricative, *semi-vowels*, and nasals. *Semi-vowels* are relatively open sounds and formed via constrictions made by the lips or tongue. See also *Fricatives, Nasals, Plosives,* and *Sibilant Fricatives*.

Sensation Level (SL): A person's *sensation level* for particular sound stimulus is calculated as a power ratio relative to their own minimum detectable level of that specific sound:

$$\text{Sensation Level} = 10\log\left(\frac{\text{Sound Intensity}}{\text{Minimum Detectable Sound Level}}\right) = \text{dB (SL)} \qquad (535)$$

Therefore if a sound is 40dB (SL) then it is 40dB above that person's minimum detectable level of the sound. Clearly the physical intensity of a *sensation level* will differ from person to person [30]. See also *Audiology, Hearing Level, Sound Level Units, Sound Pressure Level, Threshold of Hearing*.

Sensorineural Hearing Loss: If the cochlea, auditory nerve or other elements of the inner ear are not functioning correctly then the associated hearing loss is often known as *sensorineural* [30]. Typically the audiogram will reveal that the *sensorineural hearing loss* increases with increased frequency. Although a frequency selective linear amplification hearing aid will assist in some cases to reduce the impairment, in general the wearer will still have difficulty in perceiving speech signals in noisy environments. Such is the complex nature of this form of hearing loss. See also *Audiology, Audiometry, Conductive Hearing Loss, Ear, Hearing Aids, Hearing Impairment, Loudness Recruitment, Threshold of Hearing*.

Sequential Linear Feedback Register: See *Pseudo Random Binary Sequence*.

Serial Copy Management System (SCMS): The *Serial Copy Management System* provides protection from unauthorised digital copying of copyrighted material. The SCMS protocol ensures that only one digital copy is possible from a protected recording [128], [158].

Shading Weights: Coefficients used to weight the contributions of different sensors in a beamforming array (or the coefficients in an FIR filter). *Shading weights* control the characteristics of the sidelobes and mainlobe for a beamformer (or, analogously, an FIR filter). The use and design of *shading weights* is very similar to that for Data Windows and FIR filters. See also *Beamforming, Windows, FIR Filters.*

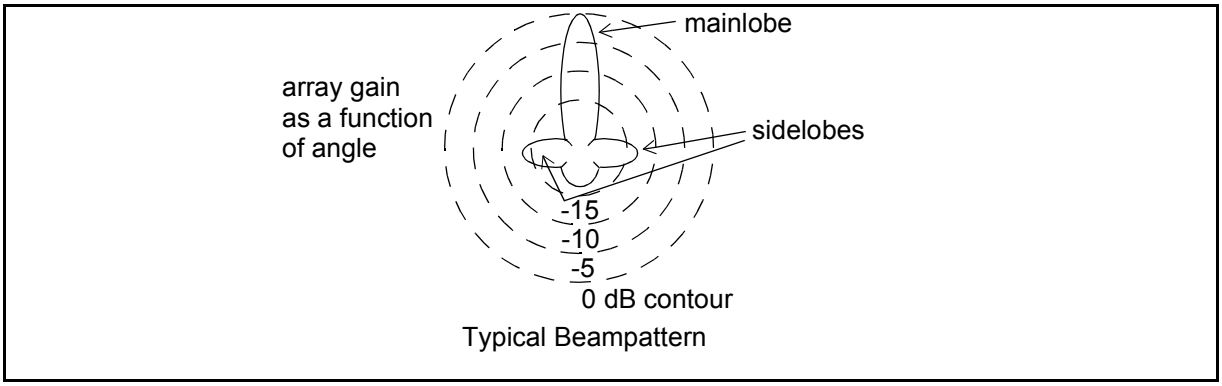
Shannon, Claude Elwood: *Claude Elwood Shannon* can be justly described as the father of the digital information age by virtue of his mathematical genius in defining the important principles of what we now call information theory. Claude Shannon was born in Michigan on April 30th 1916. He first attended University of Michigan in 1932 and graduated with a Bachelor of Science degree in Electrical Engineering, and also in Mathematics. In 1936 he joined MIT as a research assistant, and in 1938 published his first paper "A Symbolic Analysis of Relay and Switching Circuits". In 1948 he produced the celebrated paper "A Mathematical Theory of Communication" in the Bell System Technical Journal [129]. It is widely accepted that *Claude Shannon* profoundly altered virtually all aspects of communication theory and real world practice. *Claude Shannon's* other interests have included "beat the dealer" gambling machines, mirrored rooms, robot bicycle riders, and a long time interest in the practical and mathematical aspects of juggling. Readers are referred to *Shannon's* biography and collected papers [41] for more insights on this most interesting individual.

Sherman-Morrison-Woodbury Formula: See *Matrix Properties - Inversion Lemma.*

Shielded Pair: Two insulated wires in a cable wrapped with metallic braid or foil to prevent interference and provide reduced transmission noise.

Sibilant Fricatives: One of the elementary sounds of speech, namely plosives, fricatives, sibilant fricative, semi-vowels, and nasals. *Sibilant fricatives* are the hissing sounds formed when air is forced over the cutting edges of the front teeth with the lips slightly parted. See also *Fricatives, Nasals, Plosive*s, and *Semi-vowels*.

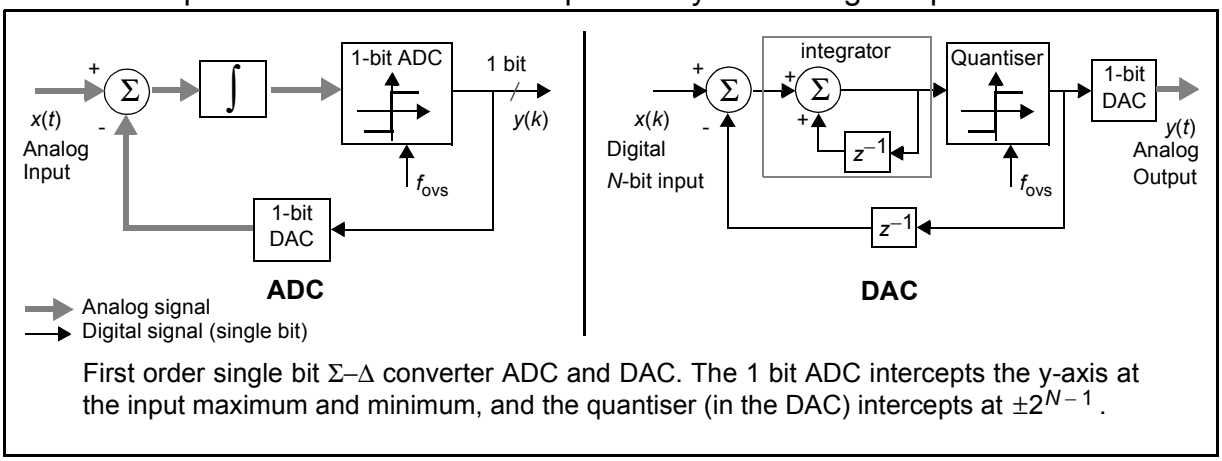
Sidelobes: In an antenna or sensor array processing system, *sidelobes* refer to the secondary lobes of sensitivity in the beampattern. For a filter or a data window, *sidelobes* refer to the stopband lobes of sensitivity. The lower the *sidelobe* level, the more selective or sensitive a given system is said to be. The level of the first *sidelobe* (relative to the main lobe peak) is often an important parameter for a data window, a digital filter, or an antenna system. *Sidelobes* are best illustrated by an example.



Typical Beampattern

See also *Main lobe, Beamformer, Beampattern, Windows.*

Sigma Delta (Σ–Δ): Σ–Δ converters use noise shaping techniques whereby the baseband quantization noise from oversampling can be high pass filtered, and the oversampling factor required to increase signal resolution can be reduced from the 4x's per single bit normally required when oversampling (see *Oversampling*). A simple first order Σ–Δ ADC converter only requires the analog components of an integrator, a summer, a 1 bit quantiser (or a 1 bit ADC) and single bit DAC in the feedback loop. A first order Σ–Δ DAC requires only the analog components of a 1 bit DAC:

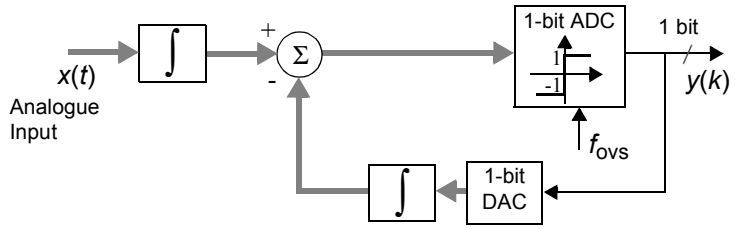


First order single bit Σ–Δ converter ADC and DAC. The 1 bit ADC intercepts the y-axis at the input maximum and minimum, and the quantiser (in the DAC) intercepts at $\pm 2^{N-1}$.

For the Σ–Δ ADC the integrator can be produced using a capacitive component, the summer using a simple summation amplifier, and the quantiser using a comparator.

Unlike conventional data converters the non linear element (the quantiser) is within a feedback loop in a mixed analogue/digital system and as a result ΣΔ devices are difficult to analyze. However as a first step to understanding the principle of operation of a ΣΔ device consider the following

representation of the ADC which is similar to the one above but the integrator has now been moved in front of the adder.



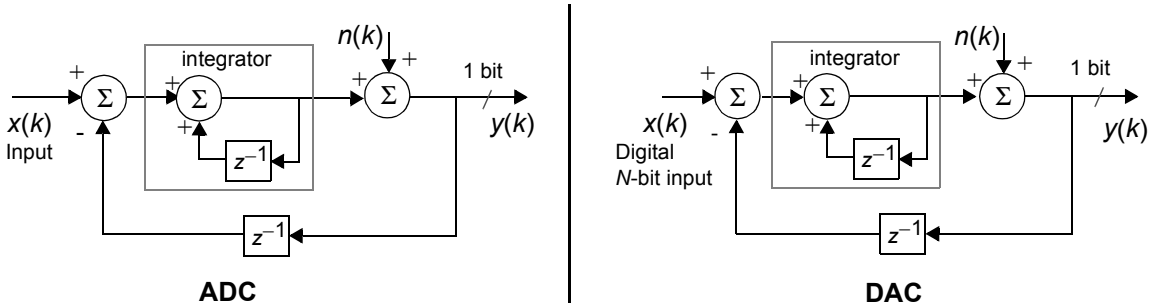
Modified first order sigma delta ADC.

Clearly the ΣΔ modulator tries to keep the mean value of the 1-bit high frequency signal equal to the mean of the input signal. Thus for a frequency input of 0 Hz, the mean output is not affected by quantisation noise. This simple result can be extended and for inputs of "very low frequency" with respect to the sampling frequency, f_{ovs}, and we conclude that the output will be a "good" representation of the input.

Because of the non-linearities present, the simple first order Σ–Δ "loop" is actually very difficult to analyze. Therefore the above linearized digital model which represents a "reasonably" mathematically tractable model is used [8]. The analog integrator is modelled with a digital integrator and the quantizer is modelled as an additive white noise source. The ADC is therefore linearised and replaced by a signal independent white noise source, $n(k)$, of variance (power) $q^2/12$ (where q is the step size of the single bit quantiser) and the analog integrator approximated by a digital integrator such that:

$$y(k) = x(k) + y(k-1) = \sum_{n=0}^{k} x(n) \approx \int_{0}^{t} x(\tau)d\tau \qquad (536)$$

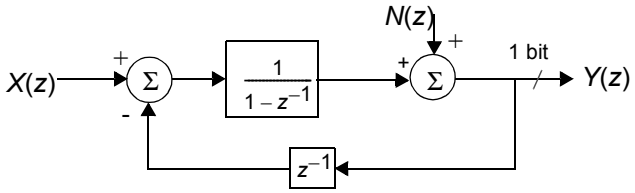
where $t = kT$ and T is the sampling period. The following analysis models are therefore realised:



ADC **DAC**

The (identical) linearised digital models for a ΣΔ ADC and a ΣΔ DAC. The linearised model allows for a more simple analysis of the behaviour of the circuits. Note that z^{-1} represents a sample delay element of period $t_{ovs} = 1/f_{ovs}$.

This z-domain model can further be simplified to:



Linearised digital models for a ΣΔ ADCs and a ΣΔ DACs. Compared to the previous figure, the integrator can be represented as a simple pole.

The output of the above ΣΔ first order model is simply given by:

$$\begin{aligned} Y(z) &= [X(z) - z^{-1}Y(z)]\left[\frac{1}{1-z^{-1}}\right] + N(z) \\ \Rightarrow Y(z) - z^{-1}Y(z) &= [X(z) - z^{-1}Y(z)] + N(z)(1-z^{-1}) \\ \Rightarrow Y(z) &= X(z) + N(z) - z^{-1}N(z) \end{aligned} \tag{537}$$

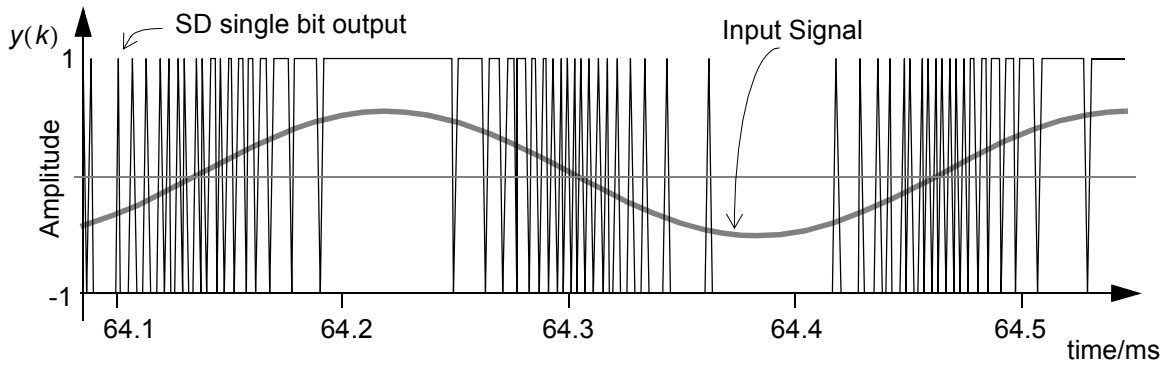
Written in the time domain the output is therefore:

$$y(k) = x(k) + n(k) - n(k-1) \tag{538}$$

From Eq. 538 we can note that the input signal passes unaltered through the modulator, whereas the added noise is high pass filtered (for low frequency values of $n(k)$, then $n(k) - n(k-1) \approx 0$). The total quantisation noise power of the 1 bit quantiser is therefore increased by using the ΣΔ loop (actually doubled or increased by 3dB), but the low frequency quantisation noise power (i.e. at the baseband) is reduced if the sampling frequency is high enough. Compared to the 1 extra bit of resolution obtained for every increase in sampling frequency by 4 for an oversampling ADC (see *Quantization Noise-Reduction by Oversampling*), the first order ΣΔ loop brings the advantage of approximately 1.5 bits of extra resolution (in the baseband) for each doubling of the sampling frequency [8].

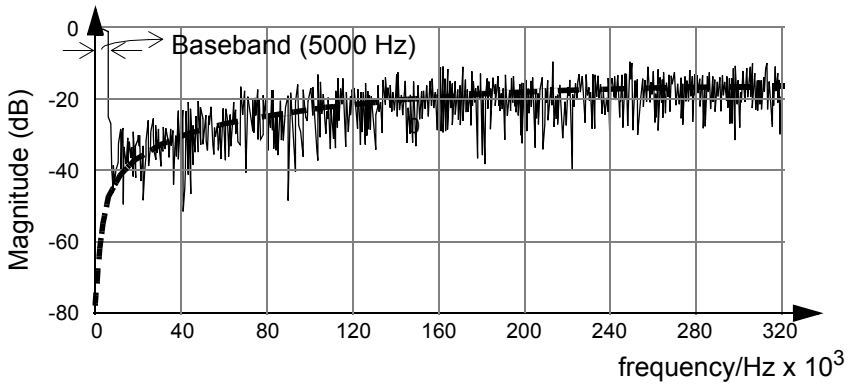
To illustrate the operation of a first order ΣΔ converter a linear chirp signal with frequency increasing from 100 to 4800 Hz over a 0.1 second interval was input to the above sigma delta loop sampling

at 64 x's the Nyquist rate, i.e. $f_{ovs} = 64 f_n = 640000$ Hz. A 0.45ms (292 samples) of the sigma delta output and chirp input input signal is shown below: .



Output of a first order sigma delta loop for a 0.45ms segement of the input chirp signal (when the signal frequency was around 3000 Hz) sampled at 640000 Hz. 292 single bit samples are shown.

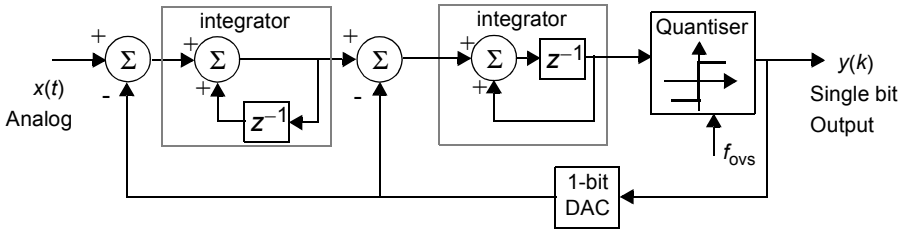
The power spectrum obtained from an FFT on about a 0.1s segment of chirp signal i.e 65536 samples (zero padded from 64000) is:



Frequency domain output of a first order, R = 64 x's oversampled sigma delta converter. The Nyquist rate was $f_s = 10000$ Hz. The input signal was a linear chirp from 100 Hz to 4800Hz over a 0.1 second interval (64000 samples) and 65536 points (≈ 0.1 seconds) were used in the (zero padded) FFT. The dotted line shows the first order noise shaping characteristic predicted by Eq. 538. By digitally low pass filtering this single bit signal, around 9-10 bits of resolution are achievable in the baseband of 0 to 5000 Hz.

Clearly the quantisation noise has been high pass filtered out of the baseband, thus giving additional resolution. The dotted line in the above figure shows the quantisation noise shaping spectrum predicted by Eq. 538. For this oversampling rate of $R = 64$ the signal to quantisation noise ratio in the baseband is about 55dB giving between 9 and 10 bits of signal resolution (cf. $20\log 2^9$ dB). If only an oversampling single bit converter was used (i.e. no $\Sigma\Delta$ loop), 64 x's oversampling would only allow about 3-4 bits of resolution. To extract the higher resolution baseband signal a low pass filter is required to extract only the baseband signal.

To obtain more than 9-10 bits resolution without further increasing the sampling frequency, a higher order sigma delta converter can be used. The circuit for a simplified second order sigma delta loop can be represented as the z-domain model:



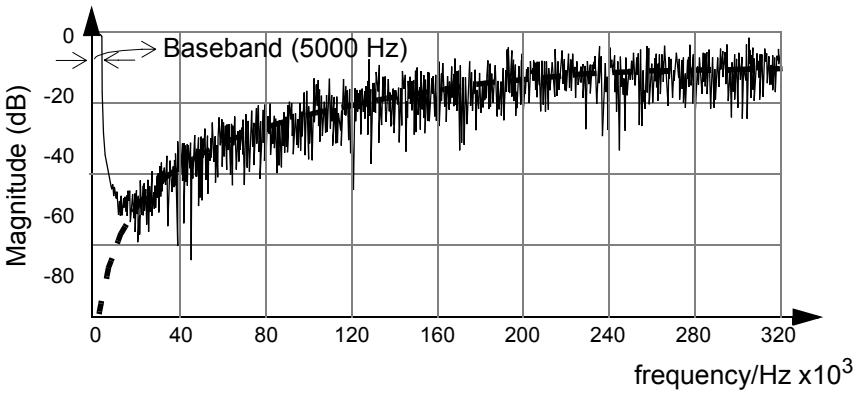
Second order sigma delta modulator. The baseband noise is much lower than that of the first order sigma delta loop due to the more effective high pass quantisation noise filtering. Analytical and experimental studies of this system are considerably more complex than that of the first order loop.

For each doubling of the sampling frequency the second order loop gives around an extra 2.5 bits resolution. The z-domain output of the above converter is:

$$Y(z) = X(z) + (1 - z^{-1})^2 N(z) \tag{539}$$

and it can be seen that this extra baseband resolution is a result of the second order high pass filtering of the quantisation noise compared to the first order loop.

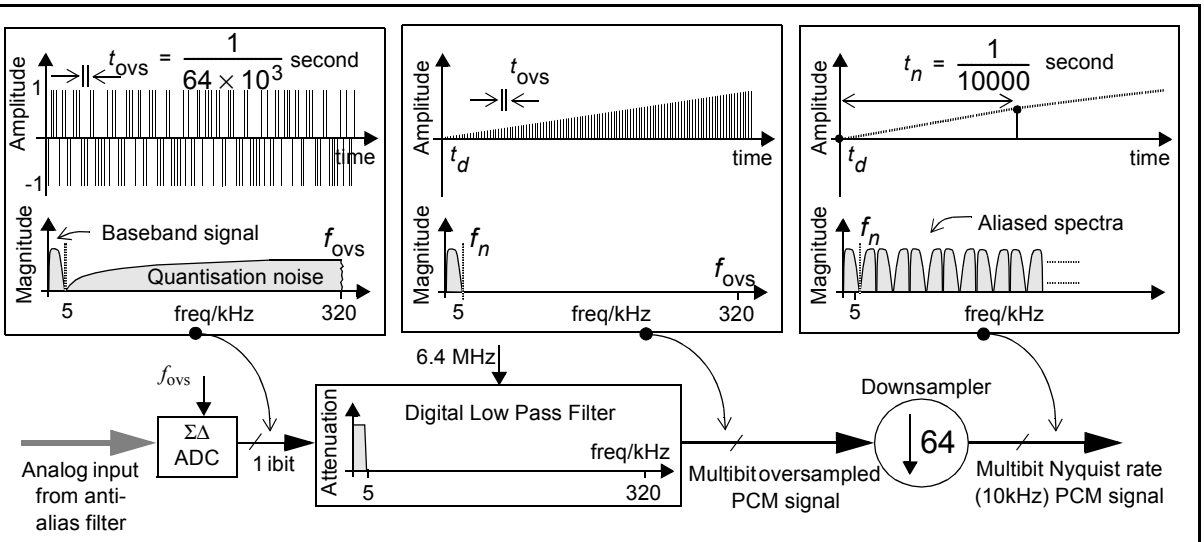
The result of inputting the same signal as previously, a linear chirp signal with frequency increasing from 100 to 4800 Hz over a 0.1 second interval at 64 x's the Nyquist rate, i.e. $f_{ovs} = 64 f_n = 640000$ Hz into a second order sigma delta modulator is:



Frequency domain output of second order $R = 64$ x's oversampled sigma delta converter. The input signal was a linear chirp signal from 100 Hz to 4800Hz over a 0.1 second interval. 65536 data points were used in the FFT. The dotted line shows the second order noise shaping characteristic predicted by Eq. 539. By digitally low pass filtering this single bit signal, around 13-14 bits of resolution are achievable in the baseband of 0 to 5000 Hz.

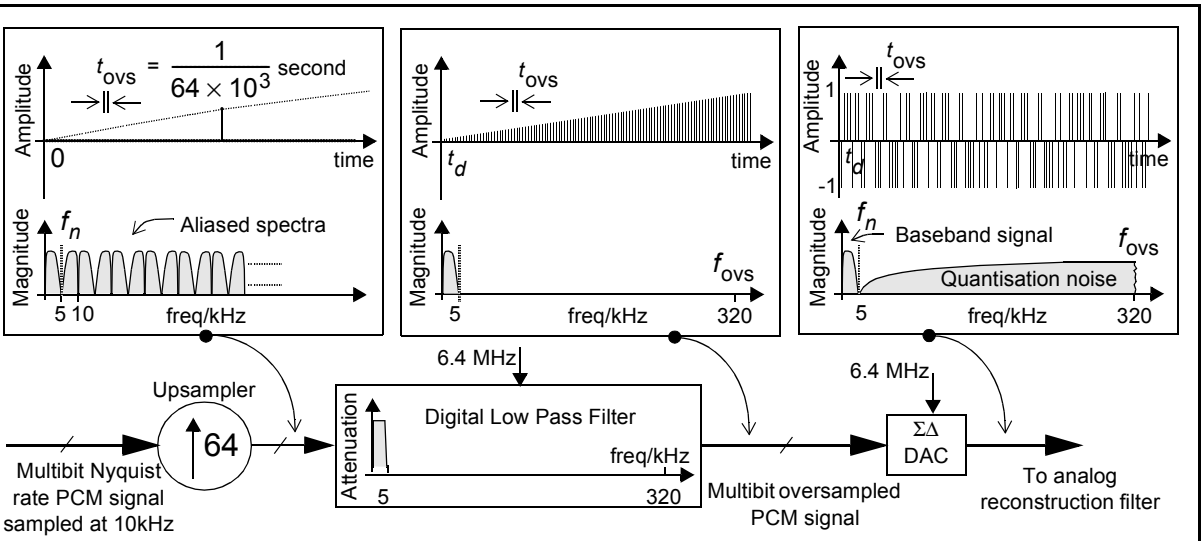
The signal to quantisation noise in the baseband is now even higher, almost of the order of 80dB and therefore allowing between 13 and 14 bits of signal resolution to be obtained (cf. $20\log 2^{13}$ dB). Note that the design of higher than second order Σ–Δ loops must be done very "carefully" in order to ensure stability and a straightforward cascading is to produced higher order loops is ill advised [8].

At the output of a Σ–Δ ADC, the single bit oversampled signal is decimated, i.e. digitally low pass filtered to half of the Nyquist frequency, and then downsampled:



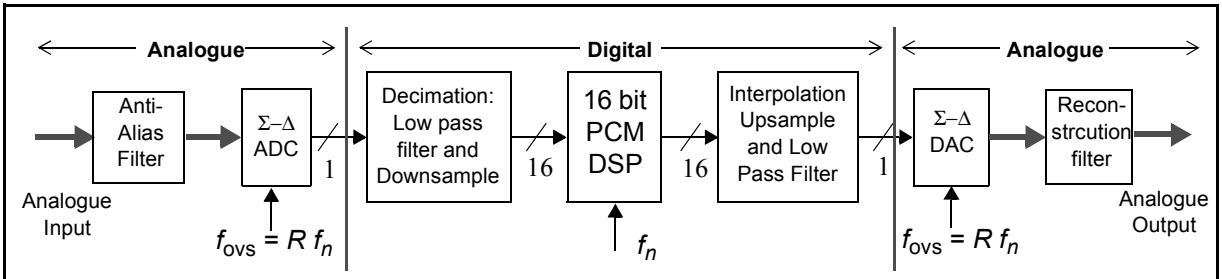
Decimation of a 64 x's oversampled sigma delta signal at $f_{ovs} = 64 \times f_n = 64$ kHz , to the Nyquist rate of $f_n = 10$ kHz by low pass digital filtering then down-sampling by 64. Note that the interpolated signal will be delayed by the group delay, t_d of the digital low pass filter (which should be linear phase in the baseband). Note that in practice the low pass filtering and downsampling is done in stages, see *Sigma Delta-Decimation*. The number of bits of signal resolution in the final output stage is a function of the order of the ΣΔ converter, and the filtering properties of the low pass filter.

In order to produce a suitably noise shaped single bit data stream for input to a ΣΔ DAC the reverse of the above process is performed:



Interpolation of a Nyquist rate signal sampled at $f_n = 10$ kHz , to a sampling rate of $64 \times f_n = 64$ kHz by upsamping and low pass digital filtering. Note that the interpolated Nyquist rate or baseband signal will be delayed by the group delay, t_d of the digital low pass filter (which should be linear phase in the baseband). Note that in practice the low pass filtering and upsampling is done in stages, see *Sigma Delta-Interpolation*. The number of bits of signal resolution in the final output stage is a function of the order of the ΣΔ converter, and the properties of the low pass filter.

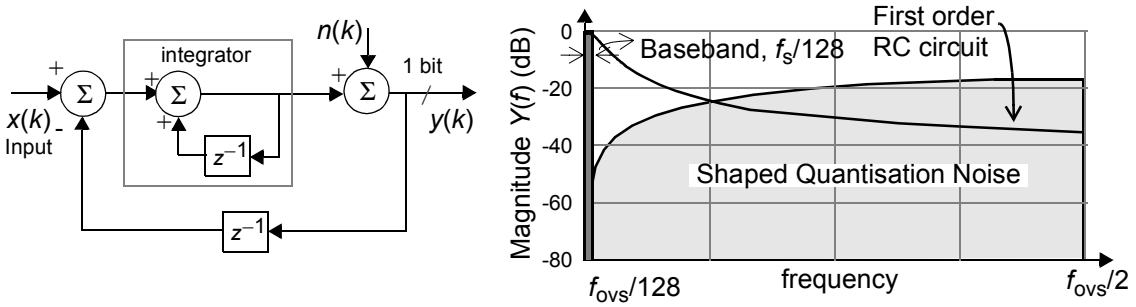
To use sigma delta converters in a DSP system computing at the Nyquist rate, the following components are required:



Using sigma delta converters as part of an DSP system. The analogue anti-alias and reconstruction filters are simple low order filters which match the order of the ΣΔ codec. The DSP processor is running at the Nyquist rate, f_n and interpolation and decimation stages are used to convert the oversampled 1 bit digital signal to a multibit Nyquist rate digital signal.

See also *Decimation, Differentiator, Integrator, Oversampling, Interpolation , Quantisation Noise - Reduction by Oversampling, Sigma Delta - Anti-Alias Filter, Sigma Delta - Decimation Filters, Sigma Delta - Reconstruction Filter.*

Sigma Delta, Anti-Alias Filter: One of the advantages of using sigma delta converters is the analogue anti-alias and reconstruction filters are very simple and therefore low cost. Consider a first order order sigma delta loop oversampling at 64 x's the Nyquist rate, with the quantiser modelled as white noise source, $n(k)$ (see *Sigma Delta*), and the input signal of full scale deflection (represented as 0dB) and occupying the entire Nyquist bandwidth:



Using the simple first order sigma delta model (left hand side), the frequency spectra shows that the quantisation noise is low in the region of the baseband, and the multibit signal representation can be extracted from the 1 bit signal by digital low pass filtering and downsampling by 64. To ensure aliasing does not occur, an analog anti-alias filter (first order RC circuit) removing frequency components above $f_{ovs}/2$ is required.

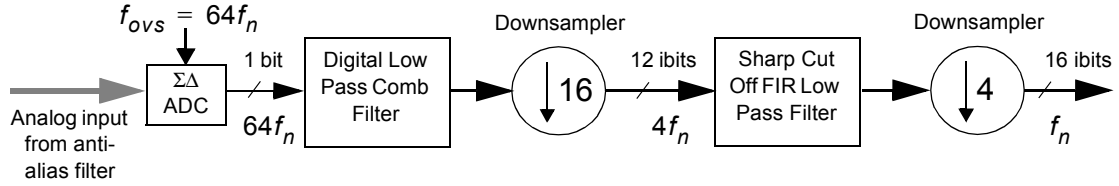
In order that aliasing does not occur, the analog anti-alias filter must cut-off all frequencies above $f_{ovs}/2$. Noting that the digital low pass decimation filter (see Sigma Delta) will filter all frequencies between $f_{ovs}/2$ and $f_{ovs}/128$, then the analog anti-alias only requires to cut off above $f_{ovs}/2$. The anti-alias filter should be cutting off by at least the baseband resolution of the converter. Therefore noting that the power roll off of an RC circuit is 6dB/octave, then if the 3dB frequency is placed at $f_{ovs}/128$, at 64 times this frequency (6 octaves) 36dB of attenuation is produced at $f_{ovs}/2$. Noting that the quantisation noise power is already about 20dB below the 0dB level at $f_{ovs}/2$, then a total of 56dB of attentuation is produced.

For a second order sigma delta converter via a similar argument as above, a second order anti-alias filter is required, (noting that the quantisation noise at $f_{ovs}/2$ is now increased due to enhanced noise

shaping). In general for an *n*-th order sigma delta converter an *n*-th order anti-alias filter should be used. The same is true for the reconstruction filter used with a sigma delta DAC. See also *Oversampling, Sigma Delta*.

Sigma Delta Converter: See *Sigma Delta*.

Sigma Delta, Decimation Filters: Decimation for a sigma delta converter requires that a low pass filter with a cut off frequency of 1/*R*-th of the oversampling frequency is implemented, where *R* is the oversampling ratio. This filter should also have linear phase in the passband. To implement a low pass FIR filter with 90dB stopband rejection and a passband of, for example, 1/64 of the sampling rate ($R = 64$) would require *thousands* of filter weights. Clearly this is impractical to implement. Therefore the low pass filtering and downsampling is often done in stages, using initial stages of simple comb type filters where all filter coefficients are of value 1 leading to a simple FIR that requires only additions and no multiplications. After this initial coarse filtering, a sharp cut-off FIR filter (still of a hundred or more weights) can be used at the final stage:



Decimation of the output of a 3rd order sigma delta converter using a low pass comb filter followed by a sharper cut off low pass FIR filter running at only 4 x's the Nyquist rate, f_n. Interpolation for a ΣΔ ADC is the effective reverse of the above process.

See also *Comb Filter, Decimation, Sigma Delta, Sigma Delta - Anti-Alias Filter*.

Sigma Delta (Σ–Δ) Loop: A term sometimes used to indicate a first order sigma delta converter. The "loop" refers to the feedback from converter output to an input summation state. See *Sigma Delta*.

Sigma Delta, Reconstruction Filter: The order of the reconstruction filter for a sigma delta DAC should match that of sigma delta order. For details see *Sigma Delta - Anti Alias Filter*.

Sign Data/Regressor LMS: See *Least Mean Squares Algorithm Variants.*

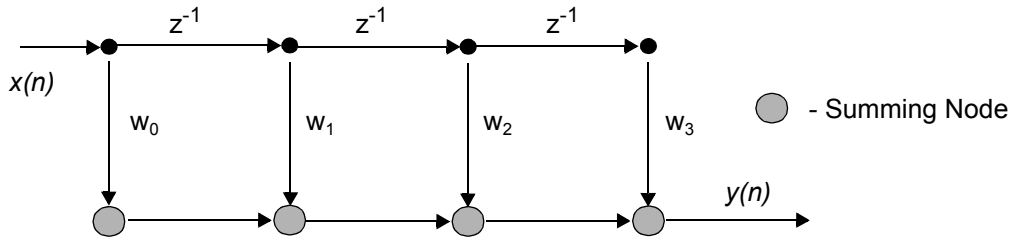
Sign Error LMS: See *Least Mean Squares Algorithm Variants.*

Sign-Sign LMS: See *Least Mean Squares Algorithm Variants.*

Signal Conditioning: The stage where a signal from a sensor is amplified (or attenuated) and anti-alias filtered in order that its peak to peak voltage, $V_{\text{Pk to Pk}}$, swing matches the voltage swing of the A/D converter and so that the signal components are not aliased upon sampling and conversion. Signals are also *conditioned* going the opposite way from D/A converter to *signal conditioning* amplifier, to actuator.

Signal Flow Graph (SFG): A simple line diagram used to illustrate the operation of an algorithm; particularly the flow of data. *Signal flow graphs* consist of annotated directed lines and splitting and summing nodes. It is very often easier to represent an algorithm in signal flow graph form than it is

to represent it algebraically. See, for example, the Fast Fourier Transform signal flow graph. Below a z-domain *signal flow graph* is illustrated for a 4 tap FIR filter.



Signal Flow Graph for a 4 tap FIR filter.

Signal Primitives: See *Elementary Signals.*

Signal Space: *Signal space* is a convenient tool for representing signals (or symbols) used for encoding information to be sent over a channel. The *signal space* approach to digital communication systems exploits the fact that a *finite* number of signals can be represented as points (or vectors) in a finite dimensional vector space. This vector space representation allows convenient matrix-vector notation (linear algebra) to be used in the design and analysis of these systems. See also *Vector Space, Matrix.*

Signal to Interference plus Noise Ratio (SINR): The ratio of the signal power to the interference power plus the noise power. Used especially in systems that experience significant interference components (e.g., intentional jamming) in addition to additive noise.

Signal to Noise Ratio (SNR, S/N): The ratio of the power of a signal to the power of contaminating (and unwanted) noise. Clearly a very high *SNR* is desirable in most systems. *SNR ratios* are usually given in dB's and calculated from:

$$\text{SNR} = 10\log_{10}\left(\frac{\text{Signal Power}}{\text{Noise Power}}\right) \tag{540}$$

Simplex: Pertaining to the ability to send data in one direction only. See also *Full Duplex, Half Duplex*.

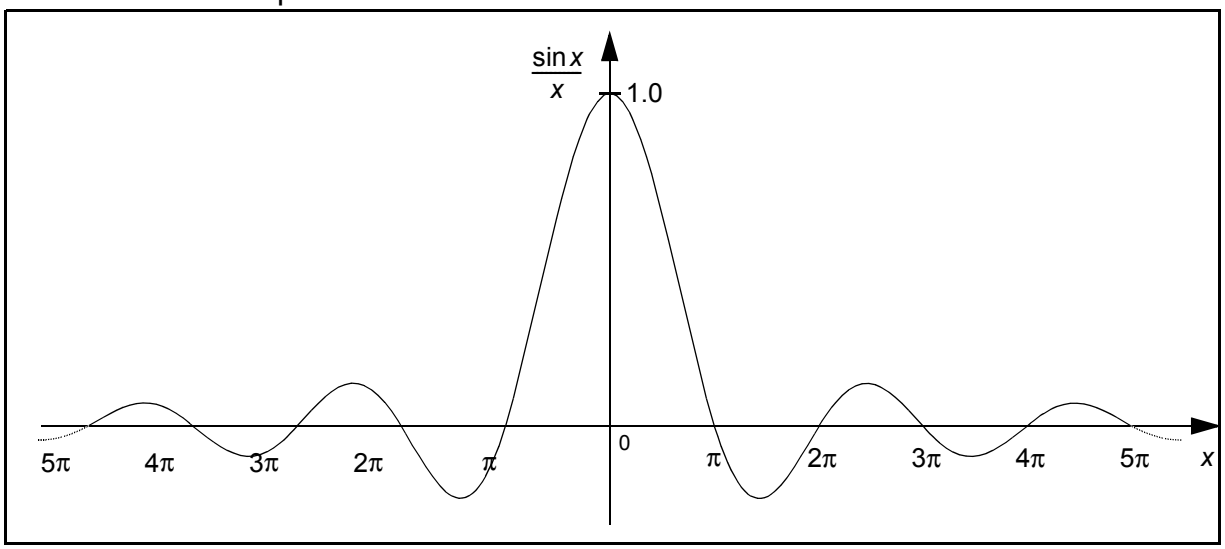
Similarity Transform: See *Matrix Decompositions - Similarity Transform*.

Simultaneous Masking: See *Spectral Masking*.

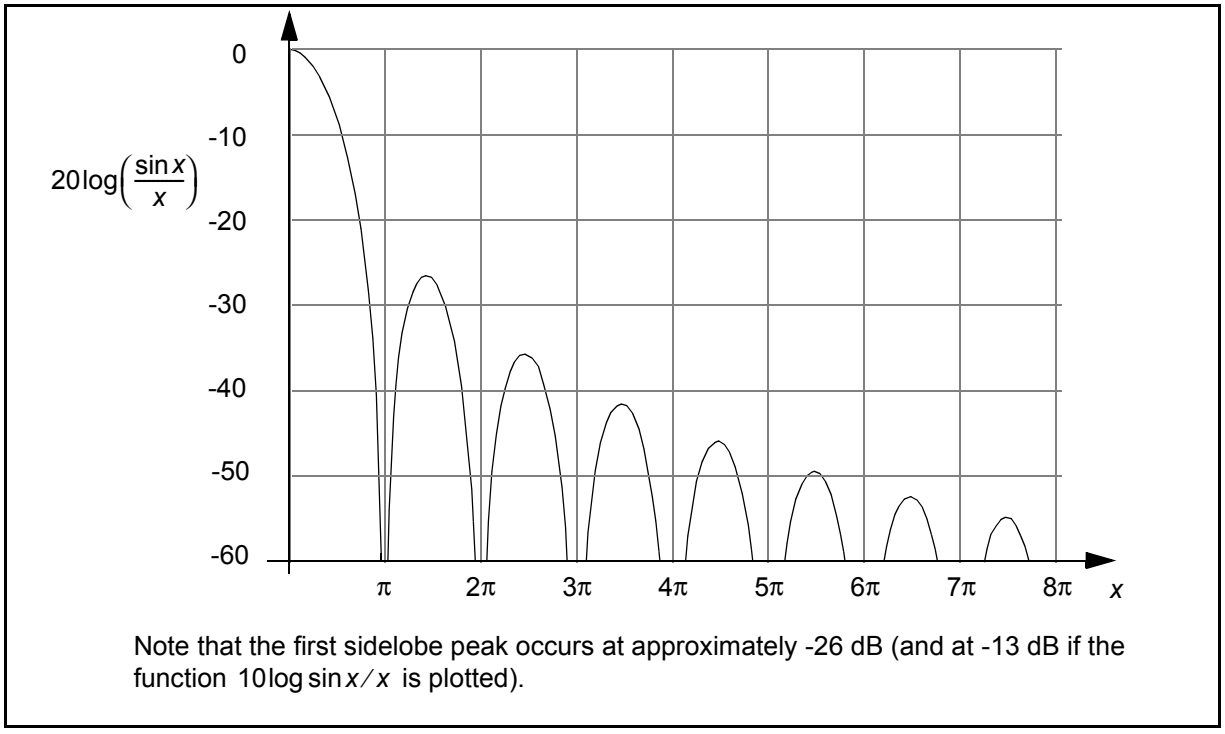
Sinc Function: The *sinc function* is widely used in signal processing and is usually denoted as:

$$\text{sinc}(x) = \frac{\sin x}{x} \tag{541}$$

The *sinc function* can be plotted as:



The logarithmic magnitude *sinc function* (which is symmetric about the y-axis) has the form:



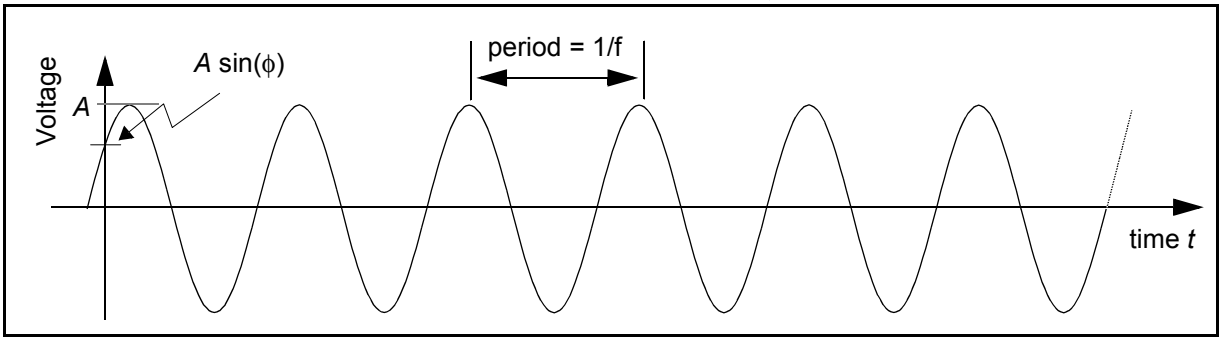
Note that the first sidelobe peak occurs at approximately -26 dB (and at -13 dB if the function $10\log \sin x/x$ is plotted).

Singular Value: See *Matrix Decompositions - Singular Value*.

Sine Wave: A *sine* wave (occurring with respect to time) can be written as:

$$x(t) = A\sin(2\pi ft + \phi) \tag{542}$$

where *A* is the signal amplitude; *f* is the frequency in Hertz; ϕ is the phase and *t* is time.



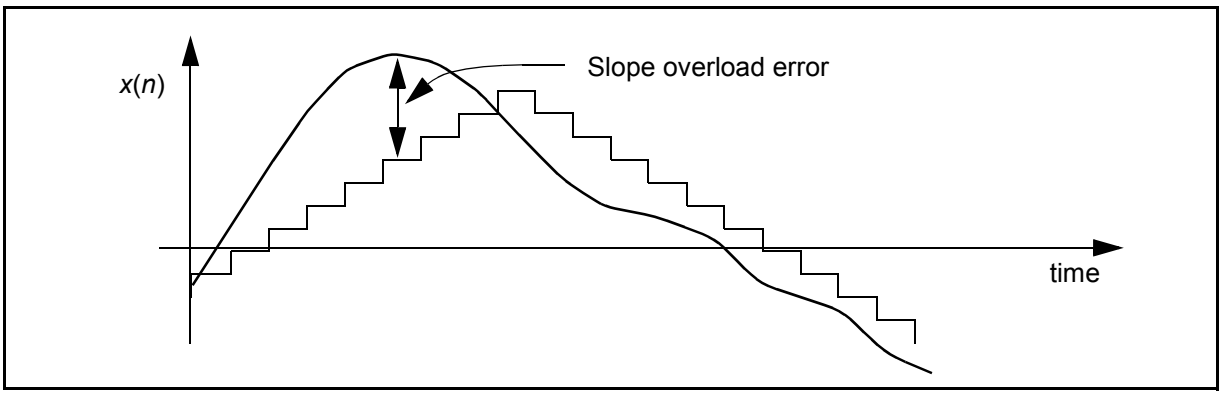
Sine Wave Generation: See *Dual Tone Multifrequency - Tone Generation*.

Single Cycle Execution: Many DSP processors can perform a full precision multiplication (e.g., 16 bit integer, 32 bit floating point - 24 bit mantissa, 8 bit exponent) and accumulate (MAC) *(a.b +c)* in a *single cycle* of the clock used to control the DSP processor. See *DSP Processor, Parallel Multiplier*.

Single Pole: If the input-output transfer function of a circuit has only one pole (in the s-domain), then it is often referred to a *single pole circuit*. The magnitude frequency plot of a *single pole circuit* will roll-off at 20dB/decade (6dB/octave). An RC circuit is a simple *single pole circuit*. See also *Active Filter, RC Circuit.*

Singular Matrix: See *Matrix Properties - Singular.*

Slope Overload: If the step size is too small when delta modulating a digital signal, then *slope overload* will occur resulting in a large error between the coded signal and the original signal. *Slope overload* can be corrected by increasing the sampling frequency, or increasing the delta (Δ) step size, although the latter may lead to granularity effects. See also *Delta Modulation, Granularity Effects* .



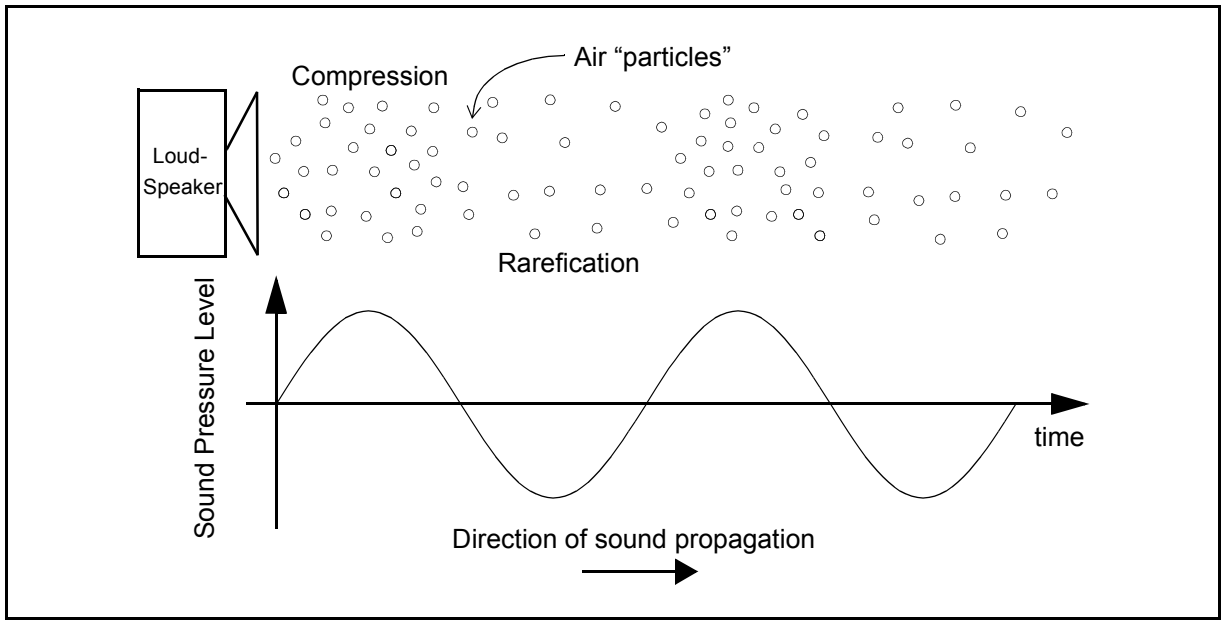
Snap-In Digital Filter: The name used to mean a digital filter that can easily be introduced between the analog front end (A/Ds) and the user interface (the PC screen). A term introduced by Hyperception Inc.

Solenoid: A device that converts electro-magnetic energy into physical displacement.

Sones: A *sone* is a subjective measure of loudness which relates the logarithmic response of the human ear to SPL. One *sone* is the level of loudness experienced by listening to a sound of 40 phon. A measure of 2 *sones* will be twice as loud, and 0.5 *sones* will be half as loud and so on. See also *Phons, Sensation Level, Sound Pressure Level.*

Sound: *Sound* is derived from vibrations which cause the propagating medium's particles (usually air) to alternately rarify and compress. For DSP purposes *sound* can be sensed by a microphone and the electrical output sent to an analog to digital converter (ADC) for input to a DSP processor. *Sound* can be reproduced in a DSP system using a loudspeaker.

When the loudspeaker below produces a tone the compression and rarefaction of air particles occurs in all directions of *sound* propagation. For illustrative purposes only the compression and rarefaction in one direction is shown:



Sound waves are longitudinal -- meaning that the wave fluctuations occur in the direction of propagation of the wave. As a point of comparison, electromagnetic waves are transversal -- meaning the variation occurs perpendicular to the direction of propagation. Hence subtle differences exist between modelling acoustic wave propagation and electromagnetic wave propagation. For example, there is no polarization phenomena for acoustic waves. See also *Audio, Microphone, Loudspeaker, Sound Pressure Level, Speed of Sound*.

Sound Exposure Meters: For persons subjected to noise at the workplace, a sound exposure meter can be worn which will average the "total" sound they are exposed to in a day, and the measurement can then be compared with national safety standards [46].

Sound Intensity: *Sound intensity* is a measure of the power of a sound over a given area. The ear of a healthy young person can hear sounds between frequencies around 1000 - 3000Hz at intensities as low 10^{-12} W/m^2 (the threshold of hearing) and as high as 1W/m^2 (just below the threshold of pain). Because of the human ear linear dynamic range of almost 1,000,000,000,000, absolute *sound intensity* is rarely quoted. Instead a logarithmic measure called sound pressure level (SPL) is calculated by measuring the *sound intensity* relative to a reference intensity of 10^{-12} W/m^2:

$$\mathrm{SPL} = 10\log\left(\frac{I}{I_{\mathrm{ref}}}\right) \mathrm{dB} \tag{543}$$

See also *Audiology, Equal Loudness Contours, Infrasound, Sound Pressure Level, Sound Pressure Level Weighting Curves, Threshold of Hearing, Ultrasound*.

Sound Intensity Meter: A *sound intensity meter* will use two or more identical microphones in order that simple beamforming techniques can be performed in an attempt to resolve the direction (as well as magnitude) of a noise. This can be important in noisy environments where there are several noise sources close together rather than a single noise source. Typically a *sound intensity meter* will consist of two precision microphones with very similar performance which are mounted a fixed distance apart. The sound intensity meter measures both the amplitude and relative phase and then calculates the noise amplitude and direction of arrival. By dividing the frequency analysis into bands, multiple sources at different frequencies and from different directions can be identified. *Sound intensity meters* usually measure noise over one third octave frequency bands. *Sound intensity meters* correspond to standard IEC 1043:1993. See also *Sound Intensity, Sound Pressure Level, Sound Pressure Level Weighting Curves* [46].

Sound Level Units: There are a number of different *units* by which *sound level* can be expressed. The human ear can hear sounds at pressures as low as 10^{-5} N/m^2 (approximately the threshold of hearing for a 1000Hz tone). Sound level can also be measured as sound intensities which specify dissipated power over area, rather than as a pressure; 2×10^{-5} N/m^2 is equivalent to 10^{-12}W/m^2. Because of the very large dynamic range of the human ear, most *sound level units* and related measurements are given on a logarithmic dB scale. See also *Audiometry, Equivalent Sound Continuous Level, Hearing Level, Phons, Sones, Sound, Sound Exposure Meters, Sound Intensity, Sound Intensity Meter, Sensation Level, Sound Pressure Level, Sound Pressure Level Weighting Curves, Threshold of Hearing*.

Sound Pressure Level (SPL): *Sound Pressure Level* (*SPL*) is specified in decibels (dB) and is calculated as the logarithm of a ratio:

$$\mathrm{SPL} = 10\log\left(\frac{I}{I_{\mathrm{ref}}}\right)\mathrm{dB} \tag{544}$$

where I is the sound intensity measured in Watts per square meter (W/m^2) and I_{ref} is the reference intensity of 10^{-12}W/m^2 which is the approximate lower threshold of hearing for a tone at 1000Hz. Alternatively (and more intuitively given the name *sound "pressure" level*) *SPL* can be expressed as a ratio of a measured sound pressure relative to a reference pressure, P_{ref}, of 2×10^{-5} N/m^2 = 20 μ Pa:

$$\mathrm{SPL} = 10\log\left(\frac{I}{I_{\mathrm{ref}}}\right) = 10\log\left(\frac{P^2}{P_{\mathrm{ref}}^2}\right) = 20\log\frac{P}{P_{\mathrm{ref}}}\ \mathrm{dB} \tag{545}$$

Intensity is proportional to the squared pressure. i.e.

$$I \propto P^2 \tag{546}$$

A logarithmic measure is used for sound because of the very large dynamic range of the human has a linear scale of intensity of more than 10^{12} and because of the logarithmic nature of hearing. Due to the nature of hearing, a 6dB increase in sound pressure level is not necessarily perceived as twice as loud. (See entry for *Sones*.)

Some approximate example *SPL*s are:

SPL (dB)	Intensity ratio I / I_{ref}	Pressure ratio P / P_{ref}	Example Sound
120	10^{12}	10^{6}	Gun-fire (Pain threshold)
100	10^{10}	10^{5}	The Rolling Stones
80	10^{8}	10^{4}	Noisy lecture theatre
60	10^{6}	10^{3}	Normal conversation
40	10^{4}	10^{2}	Low murmur in the countryside
20	10^{2}	10^{1}	Quiet recording studio
0	1	1	Threshold of human hearing
-10	$10^{-1} = 0.1$	$10^{-1/2} = 0.316$	The noise of a nearby spider walking.

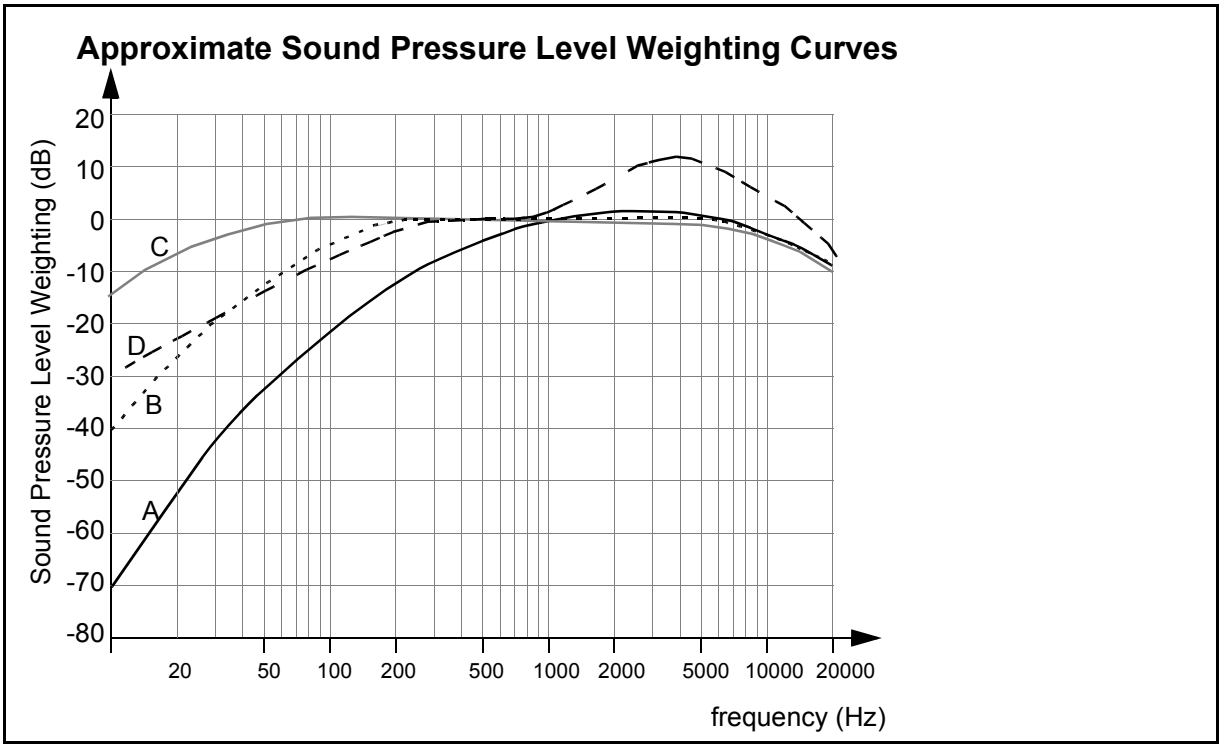
Table 2:

It is worth noting that standard atmospheric pressure is around 101300 N/m^2 and the pressure exerted by a very small insect's legs is around 10 N/m^2. Therefore the ear and other sound measuring devices are measuring extremely small variations on pressure. See also *Audiology, Audiometry, Equivalent Sound Continuous Level, Hearing Level, Sones, Sound Intensity, Sound Pressure Level, Sound Pressure Level Weighting Curves, Threshold of Hearing*.

Sound Pressure Level (SPL) Weighting Curves: Because the human ear does not perceive all frequencies of the same SPL with the same loudness, a number of *SPL weighting scales* were introduced. The most common is the A *weighting curve* (based on the average threshold of hearing) which attempts to measure acoustics signals in the same way that the ear perceives it. *Sound pressure level measurements* made using the *A-weighting curve* are indicated as dB(A) or dBA, although the use of this weighting is so widespread in SPL meters measuring environmental noise, that the A is often omitted. Sounds above 0dB(A) over the frequency range 20-16000Hz are "likely" to be perceptible by humans with unimpaired hearing. As an example of using the *weighting curve*, a 100Hz tone with *SPL* of 100dB(SPL) will register about 78dB(A) on the A-weighting scale and can be "loosely" interpreted as being 88dB above the threshold of hearing at 100Hz from the figure below.

Other less commonly used *weighting curves* are denoted as B, C and D. Standard weighting curves can be found in IEC 651: 1979, BS 5969: 1981, and ANSI S1.4-1983.

See also *Audiogram, Audiology, Hearing Level, Permanent Threshold Shift, Psychoacoustics, Sound Pressure Level, Spectral Masking, Temporal Masking, Threshold of Hearing*.



Source Coding: This refers to the coding of data bits to reduce the bit rate required to represent an information source (i.e., a bit stream). While channel coding introduces structured redundancy to allow correction and detection of channel induced errors, *source coding* attempts to reduce the natural redundancy present in any information source. The lower limit for source coding (without loss of information) is set by the entropy of the source. See also *Channel Coding, Huffman Coding, Entropy, Entropy Coding.*

Source Localization: See *Localization.*

Space: See *Vector Properties - Space.*

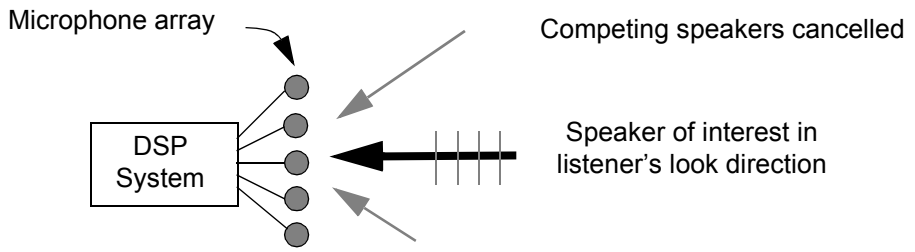
Space, Vector: See *Vector Properties and Definitions - Space.*

Span of Vectors: See *Vector Properties and Definitions - Span.*

Sparse Matrix: See *Matrix Structured - Sparse.*

Spatial Filtering: Digital filters can be used to separate signals with non-overlapping spectra in the frequency domain. A DSP system can also be set up to separate signals arriving from different

spatial locations (or directions) with an array of sensors. This process is referred to as *spatial filtering*. See *Beamforming, Beampattern.*



The DSP system identifies the broadside (head-on) waveform and attempts to null out the interfering signal from the oblique angles to produce a *spatially filtered* signal which is sent to an amplifier and a small loudspeaker in the listener's ear.

Spectral Analysis: Methods for finding the frequency content of signals, usually using the FFT and variants.

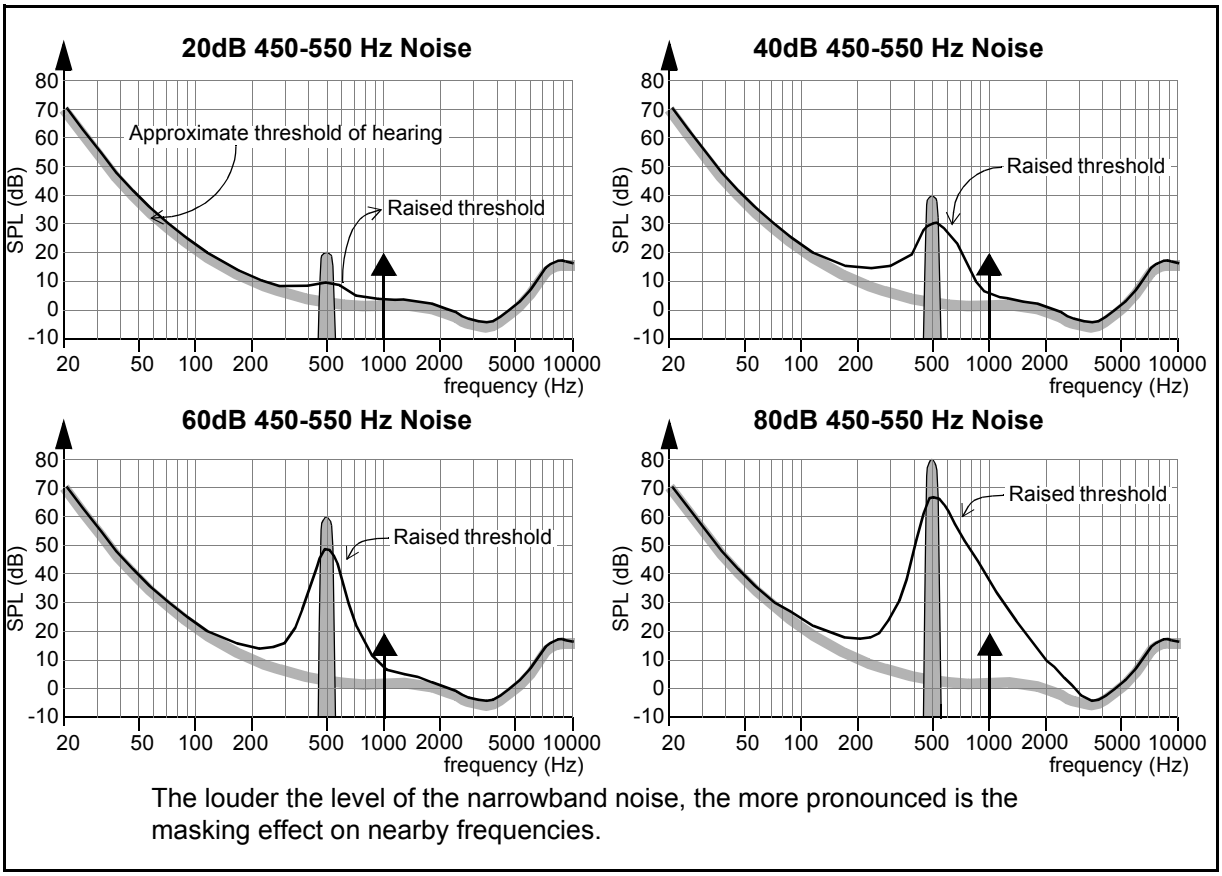
Spectral Decomposition: See *Matrix Decompositions - Spectral Decomposition*

Spectral Leakage: When a segment of data is transformed into the frequency domain using the FFT (or DFT), there will be discontinuities at the start and end of the data window unless the data window is an integral number of periods of the waveform (this is rarely the case). The discontinuities will manifest themselves in the frequency domain as sidelobes around the main peaks. *Spectral leakage* can be reduced (at the expense of wider peaks) by smoothing windows such as the Hanning, Hamming, Blackman-harris, harris, Von Hann and so on. See also *Discrete Fourier Transform - Spectral Leakage, Windows, Sidelobes.*

Spectral Masking: *Spectral masking* refers to the situation where a very loud audio signal in a certain frequency band drowns out a quieter signal of similar frequencies. A very stark example of *spectral masking* is where a conversation is rendered inaudible if standing next to a revving jet engine! *Spectral masking* is almost often referred to as simply *masking*.

Spectral masking also has more subtle and quantifiable effects whereby the presence of a signal causes the threshold of hearing of signals with a similar frequency to increase [30], [52]. For example if a narrowband of noise of approximately 100Hz bandwidth and centered at 500Hz is

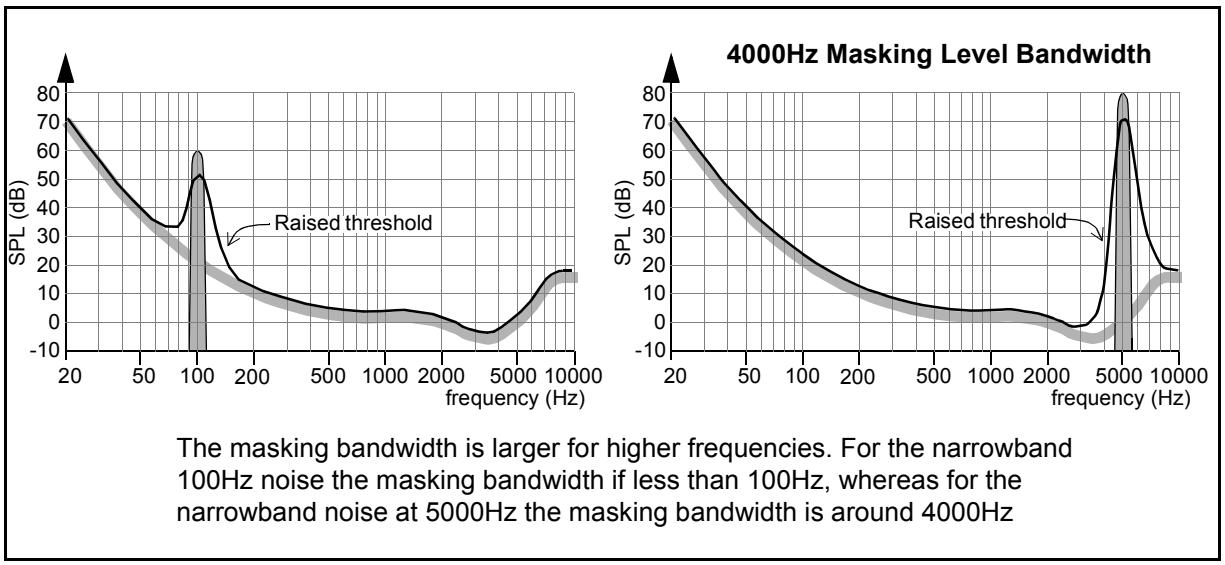
played to a listener at various different sound pressure levels, the threshold of hearing around 500Hz is raised:



The louder the level of the narrowband noise, the more pronounced is the masking effect on nearby frequencies.

The higher the SPL, the more the threshold of hearing of nearby frequencies will be raised, i.e. the more pronounced the masking effect is. In the above example when the 500Hz narrowband noise is at a level of 80dB then the 1000Hz tone at 20dB is inaudible to the human ear. In general the effect of masking is more pronounced for frequencies above the masking level. For the above example of narrowband noise, at 80dB SPL the masking effect at frequencies above 500Hz almost stretches a full octave falling off at around 60dB/octave, whereas for frequencies below 500Hz the masking effect falls off at around 120dB/octave.

The bandwidth of the masking level is higher for high frequencies. For example below 500Hz the masking level bandwidth is less that 100Hz, whereas for 10-15kHz, the bandwidth of the masking level is around 4kHz:



The masking bandwidth is larger for higher frequencies. For the narrowband 100Hz noise the masking bandwidth if less than 100Hz, whereas for the narrowband noise at 5000Hz the masking bandwidth is around 4000Hz

The auditory effects of *spectral masking* are the basis for signal compression techniques such as precision adaptive subband coding (PASC). See also *Auditory Filters, Equal Loudness Contours, Psychoacoustic subband coding (PASC), Temporal Masking, Threshold of Hearing*.

Spectrogram: A 2-D plot with time on the x-axis, and frequency on the y-axis. The magnitude at a particular frequency and a particular time on the *spectrogram* is indicated by a color (or grey scale) contour map. Widely used in speech processing.

Speech Compression: Using DSP algorithms and techniques to reduce the bit rate of speech for transmission or storage. Algorithms in wide use for communications related applications (usually speech sampled at 8kHz and 8 bit samples) that have been standardized include, LPC10, CELP, MRELP, CVSD, VSELP and so on.

Speech Immunity: Dual tone multifrequency receivers must be able to discriminate between tone pairs, and speech or other stray signals that may be present on the telephone line. The capacity of a circuit to discriminate between DTMF and other signals is often referred to at the *speech immunity*. See also *Dual Tone Multifrequency.*

Speech Processing: The use of DSP for speech coding, synthesis, or speech recognition. Speech synthesis research is more advanced, whereas speech recognition and natural language understanding continue to be a very large area of research.

Speech Recognition: Using DSP to actually interpret human speech and convert into text or trigger particular control functions (e.g. open, close and so on).

Speech Shaped Noise: If a random noise signal has similar spectral characteristics to a speech signal this may be referred to as *speech shaped noise*. Speech noise is unlikely to be intelligible and would be mainly used for DSP system testing and benchmarking. *Speech shaped noise* is also used in audiometry.

Speech Synthesis: The process of using DSP for synthesizing human speech. A simple method is to digitally record a dictionary of a few thousand commonly used words and cascade them

together to form a desired sentence. This rudimentary form of *synthesis* will have no intonation and be rather difficult to listen to and understand for long messages. It will also require a large amount of memory. True speech synthesizers can be set up with a set of formant filters, fricative formant and nasal unit and associated control algorithms (for context analysis etc.).

Speed of Sound: The *speed of sound* in air is nominally taken as being 330m/s. In actual fact, depending on the actual air pressure and temperature this speed will vary up and down. More generally the *speed of sound* will depend on the solid, liquid or gas in which it is travelling. Some typical values for the *speed of sound* are:

Substance	Approximate Speed of Sound (m/s)
Air at -10^{o}C	325
Air at 0^{o}C	330
Air at 10^{o}C	337
Air at 20^{o}C	343
Water	1500
Steel	5000-7000
Wood	3000-4000

Table 3:

See also *Absorption, Sound, Sound Pressure Level*.

SPOX: A signal processing operating system and the associated library of functions.

Spread Spectrum: *Spread spectrum* is a communication technique whereby bandwidth of the the modulated signal to be transmitted is increased, and thereafter decreased in bandwidth at the receiver [9], [16].

Square Matrix: See *Matrix Structured - Square.*

Square Root: The *square root* is a rare operation in real time DSP as most compression, digital filtering, and frequency transformation type algorithms require only multiply-accumulates with the occasional divide. *Square roots* are, however, found in some image processing routines (rotation etc) and in DSP algorithms such as QR decomposition. General purpose DSP processors do not perform *square roots* in a single cycle, as they do for multiplication, and successive approximation techniques are usually used. Consider the following iterative technique to calculate $\sqrt{a}$:

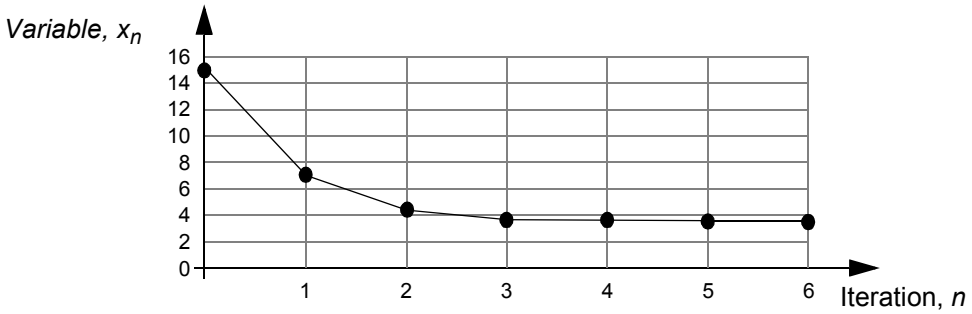
$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right) \tag{547}$$

Using an initial guess, x_0, as $a/2$ the algorithm converges asymptotically. The algorithm is often said to have converged when a specified error quantity is less than a particular value.

Finding the square root of a = 15, using the iterative update:

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right) \quad (548)$$

After only 6 iterations the algorithm has *converged* to within 0.03 of the correct solution.



Square Root Decomposition: See *Matrix Decompositions - Cholesky*.

Square Root Free Given's Rotations: See *Matrix Decompositions - Square Root Free Given's Rotations*.

Square Root Matrix: See *Matrix Properties - Square Root Matrix.*

Square System of Equations: See *Matrix Properties - Square System of Equations*.

Square Wave: A sequence of periodic rectangular pulses. See *Rectangular Pulse.*

Stability: If an algorithm in a DSP processor is stable then it is producing bounded and perhaps useful output results from the applied inputs. If an algorithm or system is not stable then it is exhibiting *instability* and outputs are likely to be oscillating. See *Instability.*

Stand-Alone DSP: Most DSP application programs are developed on DSP boards hosted by IBM PCs. After development of, for example, a DSP music effects box, the system will be *stand-alone* as it is no longer hosted by a PC.

Standards: Technology *standards* are agreed definitions, usually at the international level which allow the compatibility, reliable operation and interoperability of systems. With relevance to DSP there are various standards on telecommunications, radiocommunications, and information technology, most notable from the ISO, ITU and ETSI.

See also *Bell 103/113, Bell 202, Bell 212, Bento, Blue Book, Comité Européen de Normalisation Electrotechnique, Digital Video Interactive, European Broadcast Union, European Telecommunications Standards Institute, F-Series Recommendations, G-Series Recommendations, Global Information Infrastructure, Graphic Interchange Format, H-Series Recommendations, HyTime, I-Series Recommendations, IEEE Standard 754, Image Interchange Facility, Integrated Digital Services Network, International Electrotechnical Commission, International Mobile (Maritime) Satellite Organization, International Organisation for Standards, International Telecommunication Union, ITU-R Recommendations, ITU-T Recommendations, J-Series Recommendations, Joint Binary Image Group, Joint Photographic Experts Group, Moving*

Picture Experts Group, Multimedia and Hypermedia Information Coding Experts Group, Multipurpose Internet Mail Extensions, Multimedia Standards, Red Book, Resource Interchange File Format, T-Series Recommendations, V-Series Recommendations, X-Series Recommendations.

Static Random Access Memory (SRAM): Digital memory which can be read from or written to. *SRAM* does not need to be refreshed as does DRAM. See also *Dynamic RAM.*

Statistical Averages: See *Expected Value.*

Stationarity: See *Strict Sense Stationary, Wide Sense Stationarity*.

Status Register (SR): *See Condition Code Register*.

Step Reconstruction: See *Zero Order Hold*.

Step Size Parameter: Most adaptive algorithms require small steps while changing filter weights, parameters or signals being estimated. The size of this step is often a parameter of the algorithm called the *step size* (or the *adaptive step size*). As an example, the step size in the LMS (Least Mean Squares) algorithm is almost always denoted by μ. The larger μ, the larger the adaptive increments taken by the processor with each update. Haykin 1991, suggests a normalized LMS step size parameter, α, that is equal to μ normalized by the power of the input signal. This allows appropriate comparison of adaptive LMS processors operating with different input signals. The *step size parameter* can also vary with time -- this "variable step size" often allows adaptive algorithms to achieve faster convergence times and lower overall misadjustment simultaneously. See also *Adaptive Signal Processing, Least Mean Squares Algorithm, Least Mean Squares Algorithm Variants - Variable Step Size LMS.*

Stereo: Within DSP systems *stereo* has come to mean a system with two input channels and/or two output channels. See also *Dual, Stereophonic*.

Stereophonic: This refers to a system that has two independent audio channels. See also *Monaural, Monophonic, Binaural.*

Stochastic Conversion: If an ADC with only single bit resolution producing two levels of -1 and +1 is used, then this is often referred to as *stochastic conversion*. See also *Analog to Digital Conversion, Dithering*.

Stochastic Process: A *stochastic process* is a random process. Random signals are good examples of *stochastic processes*. A number of measurements are associated with *stochastic* signals, such as mean, variance, autocorrelation and so on. Signals such as short speech segments can be described as *stochastic*.

Stopband: The range of frequencies that are heavily attenuated by a filter. See also *Passband.*

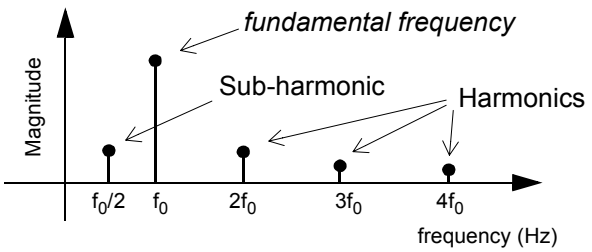
Strict Sense Stationary: A random process is strict sense stationary if it has a time invariant mean, variance, 3rd order moment and so on. For most stochastic signals, *strict stationarity* is unlikely (or difficult to show) and not (usually) a necessary criteria for analysis, modelling, etc. Usually wide sense stationarity will suffice. When texts or papers refer to a stationary process they almost always are referring to stationary in the wide sense unless explicitly stating otherwise. For DSP, particularly least mean squares type algorithms, the looser criterion of wide sense stationarity is referred to. *Strict sense stationarity* implies wide sense stationarity, but the reverse is not

necessarily true. A wide sense stationary Gaussian process, however, is also strict sense stationary. See also *Wide Sense Stationarity*.

Subband Filtering: A technique where a signal is split ino subbands and DSP algorithms are applied (usually independently) to each subband [49]. When a signal is split into subbands the sampling rate can be reduced, and very often the PCM resolution can be reduced. See also *Precision Adaptive Subband Coding*.

Subband Coding: A technique whereby a signal is filtered into frequency bands which are then coded using fewer bits than for the original wideband signal. Good *sub-band coding* schemes exist for signal compression that exploit psychoacoustic perception. See also *Precision Adaptive Subband Coding*.

Sub-Harmonic: For a given fundamental frequency produced by, for example, a vibrating string, the frequency of the harmonics are integer multiples of the fundamental frequency, and the frequency of the *subharmonics* are integer dividends of the fundamental frequency. See also *Fundamental Frequency, Harmonic. Music.*



The frequency domain representation of a fundamental frequency signal with harmonics and sub-harmonics associated harmonics.

Subspace: See *Vector Properties and Definitions - Subspace.*

Subspace, Vector: See *Vector Properties and Definitions - Subspace.*

Subtractive Synthesis: Traditional analogue technique of synthesizing music starting with a signal that contains all possible harmonics of a fundamental. Thereafter harmonic elements can be filtered out (i.e. subtracted) in order to produce the desired sound [32]. See also *Music, Western Music Scale*.

Successive Approximation: A type of A/D converter which converts from analog voltage to digital values using an approximation technique based on a D/A converter.

Super Bit Mapping (SBM): SBM (a trademark of Sony) is noise shaping FIR filter algorithm developed by Sony for mastering of compact disks from 20 bit master sources. It is essentially a noise shaping FIR filter of order 12 which produces a high pass noise shaping curve.

Surround Sound: A number of systems have been developed to create the impression that sound is spread over a wide area with the listener standing in the centre. DSP techniques are widely used to create artificial echo and reverberation to simulate the acoustics of stadiums and theatres. Dolby *Surround Sound* is widely used on the soundtracks of many major film releases. To be truly effective the sound should be coming from 360^o with loudspeakers placed at the front and back of the listener.

Sustain: See *Attack-Decay-Sustain-Release*.

Switch: A device with (typically) two states, e.g. off and on; high or low etc. Also a means of connecting/disconnecting two systems.

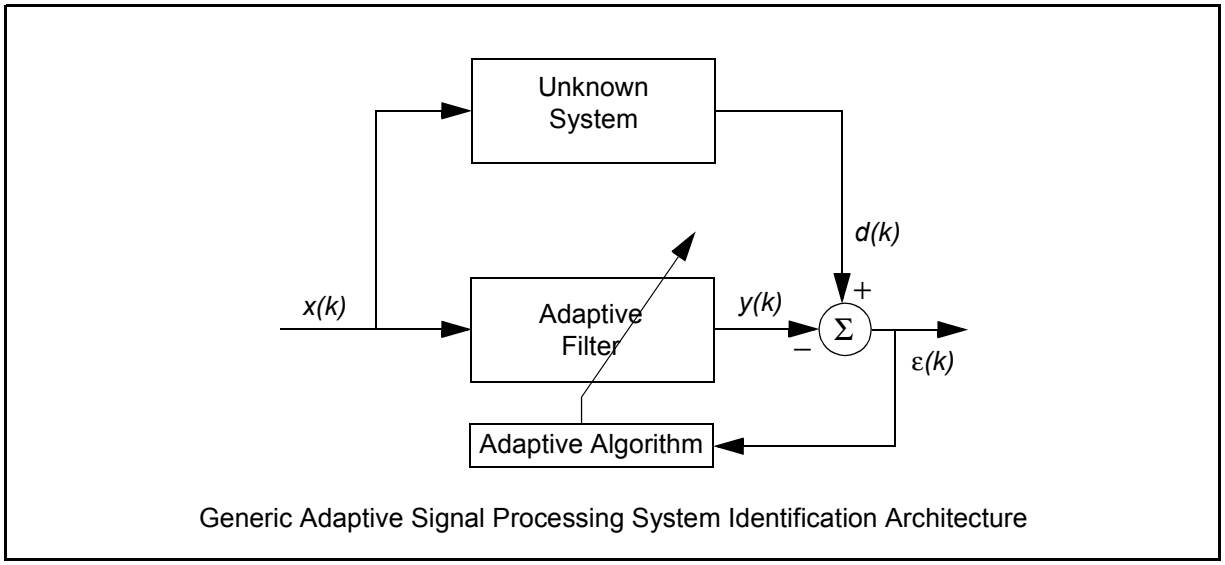
Symbol: In a digital communications system the transmission and reception of information occurs in discrete chunks. The *symbol* is the signal (one from a *finite* set) transmitted over the channel during the *symbol* period. The receiver detects which of the finite set of symbols was sent during each *symbol* period. The message is recovered by the decoding of the received *symbol* stream. The packaging of the message into discrete *symbols* sent over regular intervals forms the fundamental basis of any digital communication system. See also *Digital Communications, Message, Symbol Period.*

Symbol Period: In a digital communication system, the *symbol period* defines the regular time interval over which symbols are transmitted. During a *symbol period* exactly one of a finite number of signals are transmitted over the communications channel. Accurate knowledge of when this period begins and ends (synchronization) is required at the receiver in a communications system. See also *Symbol, Digital Communications.*

Symmetric Matrix: See *Matrix Structured - Symmetric.*

Synchronous: Meaning a system in which all transitions are regulated by a *synchronizing* clock.

System Identification: Using adaptive filtering techniques, an unknown filter or plant can be identified. In an adaptive *system identification* architecture, when the error, $\varepsilon(k)$ has adapted to a minimum value (ideally zero) then, in some sense, $y(k) \approx d(k)$, and therefore the transfer function of the adaptive filter is now similar to, or the same as, the unknown filter or system. An example application of *system identification* would be to identify the transfer function of the acoustics of a room. See also *Adaptive Filtering, Inverse System Identification, LMS algorithm, Active Noise Cancellation* .



Generic Adaptive Signal Processing System Identification Architecture

Systolic arrays: A generic name for a DSP system that consists of a large number of very simple processors interconnected to solve larger problems [25].

T

T-Series Recommendations: The T-series telecommunication recommendations from the International Telecommunication (ITU), advisory committee on telecommunications (denoted ITU-T and formerly known as CCITT) provide standards for terminal characteristics protocols for telematic services and document transmission architecture. Some of the current recommendations (http://www.itu.ch) include:

T.0	Classification of facsimile apparatus for document transmission over the public networks.
T.1	Standardization of phototelegraph apparatus.
T.2	Standardization of Group 1 facsimile apparatus for document transmission.
T.3	Standardization of Group 2 facsimile apparatus for document transmission.
T.4	Standardization of Group 3 facsimile apparatus for document transmission (+ amendment).
T.6	Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus
T.10	Document facsimile transmissions over leased telephone-type circuits.
T.10 bis	Document facsimile transmissions in the general switched telephone network.
T.11	Phototelegraph transmissions on telephone-type circuit.
T.12	Range of phototelegraph transmissions on a telephone-type circuit.
T.15	Phototelegraph transmission over combined radio and metallic circuits.
T.22	Standardized test charts for document facsimile transmissions.
T.23	Standardized colour test chart for document facsimile transmissions.
T.30	Procedures for document facsimile transmission in the general switched telephone network (+amendment).
T.35	Procedure for the allocation of CCITT defined codes for non-standard facilities.
T.42	Continuous colour representation method for facsimile.
T.50	Information technology - 7-bit coded character set for information interchange.
T.51	Latin based coded character sets for telematic services.
T.53	Character coded control functions for telematic services.
T.60	Terminal equipment for use in the teletext service.
T.62bis	Control procedures for teletext and G4 facsimile services based on X.215 and X.225.
T.64	Conformance testing procedures for the teletext.
T.65	Applicability of telematic protocols and terminal characteristics to computerized communication terminals (CCTs).
T.70	Network-independent basic transport service for the telematic services.
T.71	Link Access Protocol Balanced (LAPB) extended for half-duplex physical level facility.
T.80	Common components for image compression and communication - Basic principles.
T.81	Information technology; digital compression and coding of continuous-tone still images; requirements and guidelines.
T.82	Information technology - Coded representation of picture and audio information; progressive bi-level image compression (+T82 Correction 1).
T.83	Information technology - digital compression and coding of continuous-tone still images: compliance testing.
T.90	Characteristics and protocols for terminals for telematic services in ISDN (+ amendment).
T.100	International information exchange for interactive Videotex.
T.102	Syntax-based videotex end-to-end protocols for the circuit mode ISDN.
T.103	Syntax-based videotex end-to-end protocols for the packet mode ISDN.
T.104	Packet mode access for syntax-based videotex via PSTN.
T.105	Syntax-based videotex application layer protocol.
T.106	Framework of videotex terminal protocols.
T.122	Multipoint communication service for audiographics and audiovisual conferencing service definition.
T.123	Protocol stacks for audiographic and audiovisual teleconference applications.
T.125	Multipoint communication service protocol specification.
T.351	Imaging process of character information on facsimile apparatus.

T.390	Teletext requirements for interworking with the telex service.
T.400	Introduction to document architecture, transfer and manipulation.
T.41X/ T.42X	Information technology - Open document architecture (ODA) and interchange format.
T.431	Document transfer and manipulation (DTAM) - Services and protocols - Introduction and general principles.
T.432	Document transfer and manipulation (DTAM) services and protocols - Service definition.
T.433	Document Transfer, Access and Manipulation (DTAM) - Services and protocols - Protocol specification.
T.434	Binary file transfer format for the telematic services.
T.441	Document transfer and manipulation (DTAM) - Operational structure.
T.50X	Document application profile for the interchange of various documents.
T.510	General overview of the T.510-series.
T.521	Communication application profile BT0 for document bulk transfer based on the session service.
T.522	Communication application profile BT1 for document bulk transfer.
T.523	Communication application profile DM-1 for videotex interworking.
T.541	Operational application profile for videotex interworking.
T.561	Terminal characteristics for mixed mode (MM) of operation.
T.562	Terminal characteristics for teletext processable mode(PM.1).
T.563	Terminal characteristics for Group 4 facsimile apparatus.
T.564	Gateway characteristics for videotex interworking.
T.571	Terminal characters for the telematic file transfer within teletext service.
T.611	Programming communication interface (PCI) APPLI/COM for facsimile Group 3, facsimile Group 4, teletext, telex, e-mail and file transfer services.

For additional detail consult the appropriate standard document or contact the ITU. See also *ITU-T Recommendations, International Telecommunication Union, Standards*.

Tactile Perception: Sounds below 20Hz (infrasonic or infrasound) cannot be heard by most humans, however this low frequency infrasound can be felt *tactilely*. Some pipe organs can play notes lower than 20Hz which can enhance the overall appreciation of the rest of the music in the audible range.

Tap: The name given to a data line corresponding to a delayed version of the input signal. A tapped delay line has several points (i.e., *taps*) where delayed input samples are multiplied by the individual weights of a digital filter. The number of *taps* in a digital filter is equal to the number of weights or coefficients. For example, a particular FIR may be described as having 32 *taps* or 32 coefficients. The terms *taps* and *weights* (or coefficients) are used interchangeably -- this usage is imprecise, but we usually "know what is meant." See also *FIR filter, IIR filter, Adaptive Filter.*

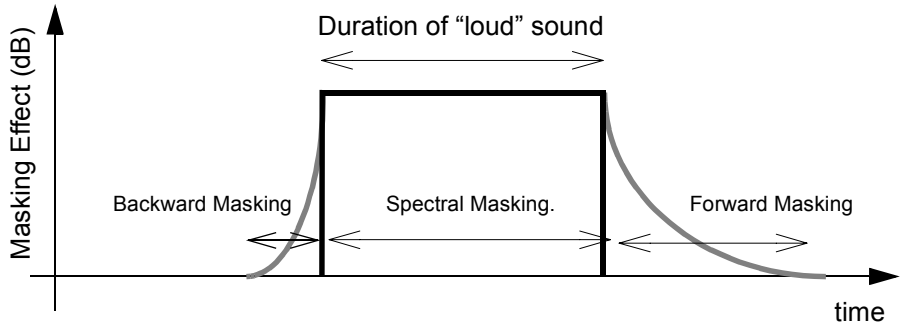
Tape Speed: See *Cassette Tape.*

Tempco: See *Temperature coefficient*.

Temperature Coefficient: The *temperature coefficient* gives a measure of the voltage (or current) drift of a component with respect to temperature change. For example if a particular 20 bit ADC (range of had a temperature coefficient of 1ppm/^{o}C, then this means that for a change in temperature of 1^{o}C, the output of the ADC would drift by less than 1 bit ($2^{20} = 1,048,576$).

Temporal Masking: The human ear may not perceive quiet sounds which occur a short time before or after a louder sound. This masking effect is called *temporal masking*. When the quiet sound occurs just after the louder sound (forward *temporal masking*) it may be interpreted that the ear has not "recovered" from the louder sound. If the quiet sound comes just before the louder sound then backward *temporal masking* may occur; a simple interpretation of this effect is less obvious. The effects of temporal masking are still a topic of debate and research [30].

For forward *temporal masking*, the closer together the loud and quiet sound, then the more of a masking effect that is likely to be present. The amount of masking is influenced by the frequency and sound pressure levels of the two sounds, and masking effects may occur for up to 200ms. Temporal masking can be useful for perceptual coding of audio whereby the first few milliseconds of sounds (such as after loud drumbeats) are not fully coded.



Sounds occurring just after the loud sound may in fact be (forward) masked (i.e. rendered perceptually inaudible) to the listener. A less pronounced backward masking effect also occurs.

See also *Audiology, Audiometer, Binaural Unmasking, Moving Picture Experts Group - Audio, Psychoacoustics, Psychoacoustic Subband Coding (PASC), Sound Pressure Level, Spectral Masking, Temporary Threshold Shift, Threshold of Hearing*

Temporary Threshold Shift (TTS): When the threshold of hearing is raised temporarily (i.e., the threshold eventually returns to normal) due to exposure to excessive noise a *temporary threshold shift* is said to have occurred. Recovery can be within a few minutes or take several hours. Many people have experienced this effect by attending a loud concert or shooting a gun. See also *Audiology, Audiometry, Threshold of Hearing, Permanent Threshold Shift.*

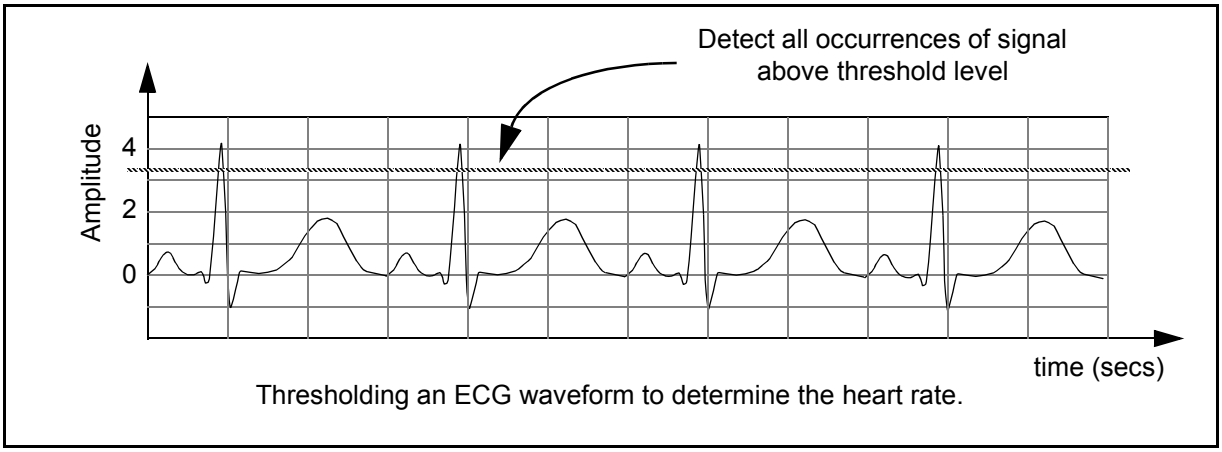
Terrestrial Broadcast: TV and radio signals are sent to consumers in one of three ways: *terrestrial*, satellite, or cable. *Terrestrial* broadcasts transmit electromagnetic waves modulated with the radio or TV signal from earth based transmitters, and are received by earth based aerials or antennas.

Third Octave Band: A typical bandwidth measure used when making measurements of sound intensity over a few octaves of frequency. The third of an octave is usually one third of the particular octave. For example, choosing octaves frequencies at 125, 250, 500Hz and so on, the bandwidths of the third of an octave bands are approximately 42Hz, 86Hz, and 166Hz. To compute a third octave frequency band around frequency f_0, note that from $2^{1/6}f_0$ down to $2^{-1/6}f_0$, the ratio of the high and low frequencies is $2^{1/3}$, or one-third of an octave (a doubling). The third octave bandwidth is computed as $(2^{1/6} - 2^{-1/6})f_0$. Three consecutive third octaves make an octave.

Third Order: Usually meaning three of a particular device cascaded together. Used in a non-consistent way. See also *Second Order.*

Threshold Detection: One of the most rudimentary forms of signal analysis, where a particular signal is monitored to find at what points it has a magnitude larger than some predefined *threshold*.

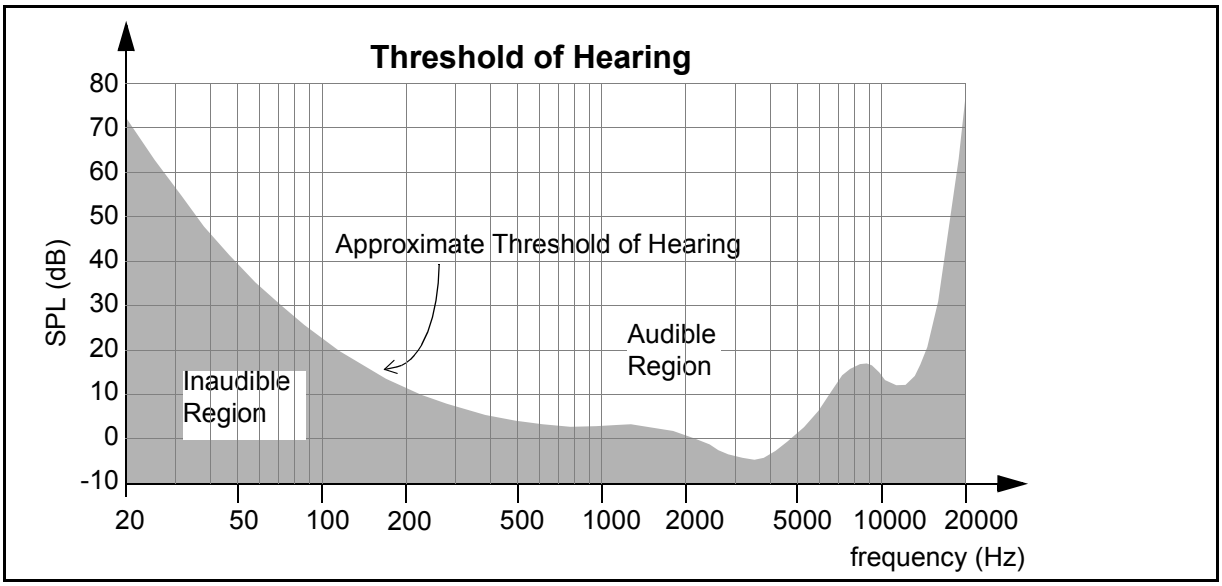
For example an ECG signal may be monitored using *threshold detection* in order to calculate the heart rate (the inverse of the R to R time).



Thresholding an ECG waveform to determine the heart rate.

Threshold of Audibility: The level of a tone that is just audible defines the *threshold of audibility* for that frequency. For a more general sound, the *threshold of audibility* is the level at which it becomes just audible. See also *Audiogram.*

Threshold of Hearing: The *threshold of hearing* or minimum audible field (MAF) is a curve of the minimum detectable sound pressure level (SPL) of pure frequency tones plotted against frequency. There are a number of different methods for obtaining the lower *threshold of hearing* depending on the actual point on/in the ear where SPL is measured, whether headphones or loudspeakers were used, and of course the cross section of population over which the averaged curve is obtained, i.e. different age groups, including/excluding hearing impaired persons and so on. (Note that although SPL was originally defined as a sound pressure level relative to the minimum detectable 1000Hz tone, established at 10^{-12}W/m^2, the average *threshold of hearing* at 1000Hz is actually around 5dB.)



The curve shown above is based on the Fletcher-Munson [73] and Robinson-Dadson [126] curves and is now a well established shape showing clearly that the ear is most sensitive to the range 1000-5000Hz where speech is found. At very low and very high frequencies the minimum thresholds increase rapidly. It is worthwhile noting that the threshold of pain is around 120dB, and prolonged exposure to such high intensities will damage the ear. The upper frequency limit of

hearing can be as high as 20kHz for very young children, but in adults is about 12-15kHz. The lower limit of hearing is often quoted as 20Hz as further reduction on frequency is not perceived as a further reduction in pitch. Also at these frequencies high SPL sounds can be “felt” as well as heard [30]. Many animals have hearing ranges well above 20kHz, the most noted example being dogs who respond to the sound made from dog whistles which humans cannot hear.

Given that the bandwidth of hi-fidelity digital audio systems is up to 22.05kHz for CD and 24kHz for DAT it would appear that the full range of hearing is more than covered. However this is one of the key issues of the CD-analogue records debate. The argument of some analog purists is that although humans cannot perceive individual tones above 20kHz, when listening to musical instruments which produce harmonic frequencies above the human range of hearing these high frequencies are perceived in some “collective” fashion. This adds to the perception of live music; the debate will doubtless continue into the next century.

See also *Audiogram, Audiometry, Auditory Filters, Binaural Unmasking, Ear, Equal Loudness Contours, Equivalent Sound Continuous Level, Frequency Range of Hearing, Habituation, Hearing Aids, Hearing Impairment, Hearing Level, Infrasound, Permanent Threshold Shift, Psychoacoustics, Sensation Level, Sound Pressure Level (SPL), Spectral Masking, Temporal Masking, Temporary Threshold Shift (TTS), Ultrasound*.

Timbre: (Pronounced *tam-ber*). The characteristic sound that distinguishes one musical instrument from another. Key components of timbre are the signal amplitude envelope and the harmonic content of the signal [14]. See also *Attack-Decay-Sustain-Release, Music, Western Music Scale*.

Time Invariant: A quantity that is constant over time. For example if the mean of a stochastic signal is described as being time invariant, then this means that the measured value of the mean will be the same if measured today, and then tomorrow.

TMS320: The part number prefix for Texas Instruments series of DSP processors. One of the early members of the family was the TMS320C10 in 1984.

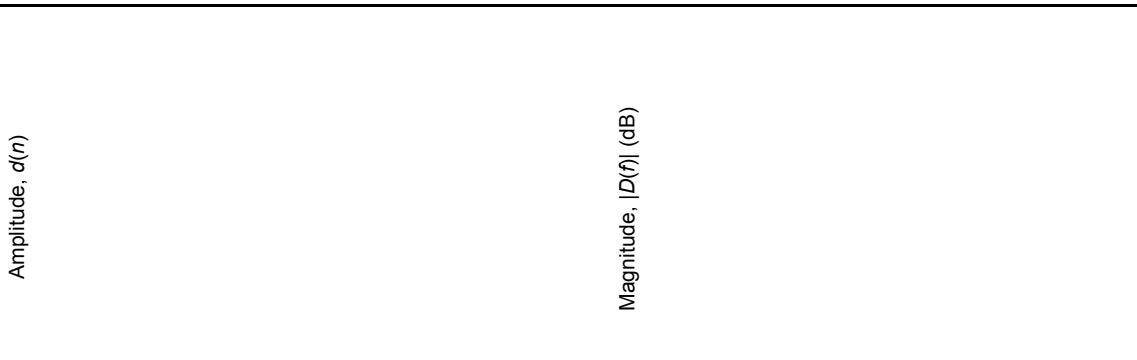
Toeplitz Matrix: See *Matrix Structured - Toeplitz.*

Tonal Distortion: If an analogue signal with periodic or quasi-periodic components is converted to a digital signal and the output contains harmonics of the periodic signal that were not present in the original, then this is referred to as tonal or harmonic distortion. For example, the following digital

signal is a 200Hz sine wave sampled at 48000Hz with an amplitude of 250. A 16384 point FFT confirms that signal there is no *tonal distortion* present.

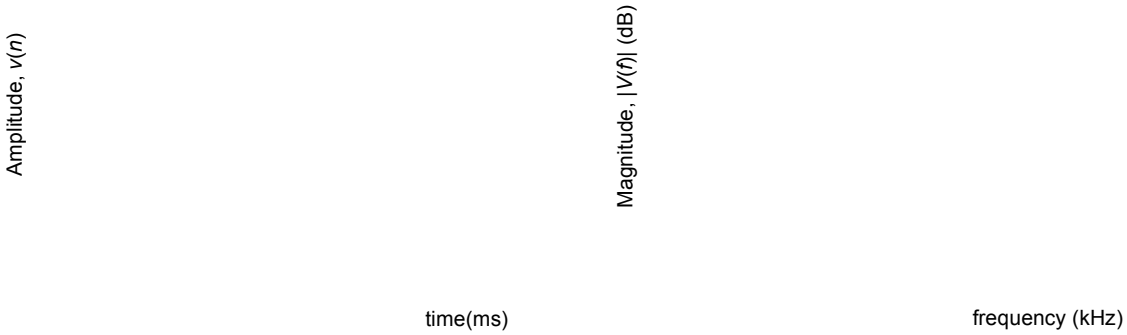
The time and frequency representations of a 200Hz sine wave of amplitude 100, sampled at 48000 Hz, i.e. $y(k) = 100\sin(((2\pi 200)k)/48000)$ The 16384 point FFT shows that there is no tonal distortion. Note that on the frequency graph an amplitude of 100 corresponds to about -50 dB ($= 20\log(50/32767)$) where the full scale amplitude of 32767 ($= 2^{15} - 1$) is 0dB

However when the signal is clipped at an amplitude of 80, then this non-linear operation causes tonal *distortion* as can be seen in the frequency domain representation:



The time and frequency representations of a 200Hz sine wave of amplitude 100, sampled at 48000 Hz, i.e. $d(k) = 100\sin(((2\pi 200)k)/48000)$ which has been clipped at ±80. The 16384 point FFT shows that there is clearly *tonal distortion* at integer multiples of the signal frequency.

Also when a very low level periodic signal is converted from an analog to a digital representation, the quantisation error will be correlated with the signal which will manifest itself as *tonal distortion*:



The time and frequency representations of a 100Hz sine wave of amplitude 5, sampled at 48000 Hz, i.e. $v(k) = 100\sin((2\pi 200)k/48000)$. The 16384 point FFT shows that there is clearly tonal distortion present.

When a speech or music signal is converted from analog to digital then the quasi-periodic nature of the signals may result in *tonal distortion* components. This *tonal distortion* may be due either to non-linearities in the system or analog-to-digital conversion of very low level signals, See also *Dithering*, *Total Harmonic Distortion*.

Tone (1): A pure sine wave (existing for all time*, t*).

Tone (2): In music theory each adjacent note in the chromatic scale differs by one semitone, which corresponds to multiplying the lower frequency by the twelfth root of 2, i.e. $2^{1/12} = 1.0594631\ldots$. A difference of two semitones is a *tone*. Coincidentally (or perhaps by design!) "tone" is an anagram of "note", as in musical note. See also *Western Music Scale*.

Tone Generation: See *Dual Tone Multifrequency - Tone Generation*.

Total Error Budget: Virtually every component in a standard input/output DSP system will contribute some error, or noise to a signal passing through. If a designer knows the tolerable error in the final system output, then from this *total error budget,* tolerances and allowable errors can be assigned to components. In a DSP system the designer will need to consider both analog and digital components in the *total error budget*.

Total Harmonic Distortion (THD): If a pure tone signal of *M* Hz is played into a system and the output is found to contain not only the original signal, but also small components at harmonic frequencies of *2M*, *3M*, and so on then *distortion* has occurred. The *THD* is calculated as the percentage of total energy contained in the harmonics to the energy of the signal itself. *THD* is usually expressed in dB. See also *Total Harmonic Distortion plus Noise*.

Total Harmonic Distortion plus Noise (THD+N): A measure often associated with ADCs and DACs defining the ratio of all spectral components over the specified bandwidth, excluding the input signal, to the rms value of the signal. See also *Total Harmonic Distortion*.

TP Algorithm: The *Turning Point algorithm* was a technique to reduce the sampling frequency of an ECG signal from 200 to 100 samples/sec. The algorithm developed from the observation that except for the QRS portion of the ECG with large amplitudes and slopes, a sampling rate of 100

samples was more than adequate. The algorithm processes three points at once in order to identify where a significant turning point occurs.

Trace of a Matrix: See *Matrix Properties - Trace.*

Transceiver: A data communications device that can both transmit and receive data.

Transcoding: Converting from one form of coded information to another. For example converting from MPEG1 compressed video to H.261 compressed video can be termed as *transcoding*.

Transducer: A device for converting one form of energy into another, e.g. a microphone converts sound energy into electrical energy.

Transform Coding: For some signals, mathematical transformation of the data into another domain may yield a data set that is more amenable to compression techniques that the original signal. The transform is usually applied to small blocks of data which are compared with a standard set of blocks to produce a correlation function for each. The signal is decompressed by applying the correlation functions as a weighting to each standard block. It is possible to combine *transform coding* and predictive coding to yield powerful compression algorithms. The disadvantage is that the algorithms are computation intensive. See also *JPEG, MPEG, DCT.*

Transfer Function: A description (usually in the mathematical Z-domain) of the function a particular linear system will perform on signals. For example, the transfer function of a very simple low pass filter, $y(n) = x(n) + x(n-1)$, could be given as the transfer function *H(z)*:

$$H(z) = \frac{Y(z)}{X(Z)} = 1 + z^{-1} \tag{549}$$

See also *Impulse Response.*

Transients: When an impulse is applied to a system, the resulting signal is often referred to as a *transient*. For example when a piano key is struck, the piano wire creates a *transient* as it continues to vibrate long after the key was struck.

Sometimes, unexplained small currents and voltages within a system are described (and perhaps dismissed) as *transients*.

Transpose Matrix: See *Matrix Operations - Transpose.*

Transpose Vector: See *Vector Properties and Definitions - Transpose.*

Transputer: A microprocessor designed by INMOS Ltd. The first and original parallel processing chips (T212, T414, and T800) had four serial links to allow intercommunication with other *Transputers*. Since its launch in 1984 the *Transputer,* despite its catchy name, failed to set the computing world on fire. Although the *Transputer* was used for many DSP applications, its slow arithmetic restricted its use and it never became a general purpose DSP.

Trellis Coded Modulation (TCM): TCM is a digital modulation technique that combines convolutional coding and decoding techniques (including the Viterbi algorithm) with signal design to reduce transmission errors in a digital communication system while retaining the same average symbol energy and system bandwidth. TCM increases the number of signals in a signal set by some factor of two without increasing the signal space dimension (i.e., the system bandwidth). The coder

and decoder exploit the increase in the number signals by separating signals both by Euclidean distance in signal space as well as free distance in the convolutional code trellis. The Viterbi algorithm is used with a Euclidean distance rather than a Hamming distance as the appropriate metric to minimize probability of error (for the additive white gaussian noise channel). Trellis Codes are often referred to as Ungerboeck Codes, after G. Ungerboeck who is credited with their development. See also *Viterbi Algorithm, Euclidean Distance, Hamming Distance.*

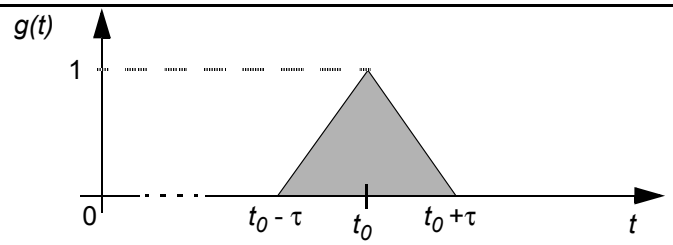
Tremolo: Tremolo is the effect where a low frequency amplitude modulation is applied to the musical output of an instrument. Tremolo can be performed digitally using simple multiplicative DSP techniques [32]:

$$\text{Tremolo Signal} = \cos(2\pi(f_t/f_s)k)s(k) \tag{550}$$

where, f_s is the sampling frequency, f_t is tremolo frequency of modulation and $s(k)$ is the original digital music signal. In practice however the tremolo effect may require more subtle forms of modulation to produce an aesthetic sound. See also *Music, Vibrato*.

Triangular Pulse (Continuous and Discrete Time): The ***continuous time triangular pulse*** can be defined as:

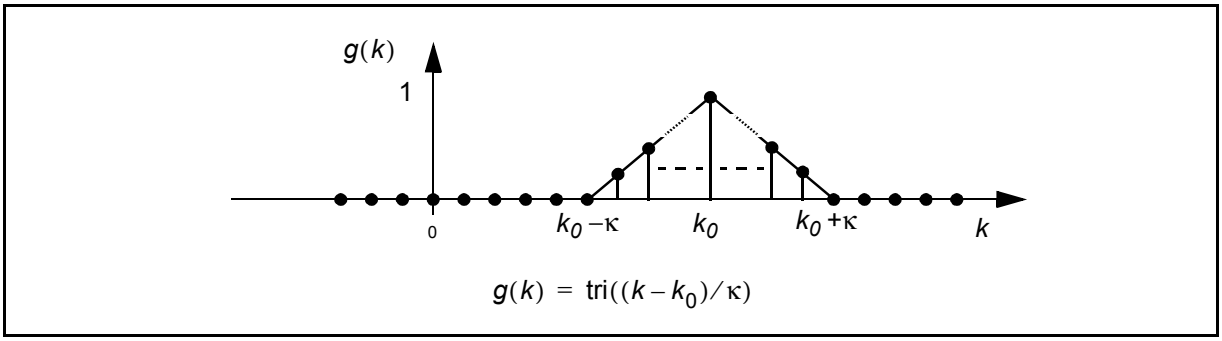
$$\text{tri}((t-t_0)/\tau) = \begin{cases} 1 - \left|\dfrac{t-t_0}{\tau}\right| & \text{if } |t-t_0| \le \tau \\ 0 & \text{otherwise} \end{cases} \quad \text{continuous time} \tag{551}$$



The *continuous triangular pulse* $g(t) = \text{tri}((t-t_0)/\tau)$

The ***discrete time triangular pulse*** can be defined as:

$$\text{tri}((k-k_0)/\kappa) = \begin{cases} 1 - \left|\dfrac{k-k_0}{\kappa}\right| & \text{if } |k-k_0| \le \kappa \\ 0 & \text{otherwise} \end{cases} \qquad \text{discrete time} \qquad (552)$$



See also *Elementary Signals, Rectangular Pulse, Square Wave, Unit Impulse Function, Unit Step Function*.

Triangularization: See *Matrix Decompositions - Cholesky/LU/QR*.

Tridiagonal Matrix: See *Matrix Structured - Tridiagonal.*

Truncation Error: When two *N* bit numbers are multiplied together, the result is a number with 2*N* bits. If a fixed point DSP processor with *N* bits resolution is used, the 2*N* bit number cannot be accommodated for future computations which can operate on only *N* bit operands. Therefore, if we assume that the original *N* bit numbers were both constrained to be less than 1 in magnitude by using a binary point, then the 2*N* bit result is also less that 1. Hence if we throw away the last *N* bits, then this is equivalent to losing precision. This loss of precision is referred to as *truncation error*. Although the *truncation error* for a single computation is usually not significant, many errors added together can be significant. Furthermore if the result of a computation yields the value of 0 (zero) after *truncation*, and this result is to be used as a divisor, a divide by zero error will occur. See also *Round-Off Error, Fractional Binary*.

Binary	0.1101011 x	0.1000100	= 0.011100011011000	⟶ Truncation	0.0111000
Decimal	0.8359375 x	0.53125	= 0.444091796875	⟶	0.4375

After multiplication of two 8 bit numbers the 16 bit result is truncated to 8 bits introducing a binary round off error of 0.000000011011000 which in decimal is 0.006591796875. If rounding had been used, then the result would have been 0.0111001, which is an error of 0.000000000101000, and in decimal an error of 0.001220703125.

Truncation Noise: When truncation errors are considered in terms of their mean power, this results in a measure of the truncation noise. See also *Truncation Error.*

Tweeter: The section of a loudspeaker that reproduces high frequencies is often called the tweeter. The name is derived from the high pitched *tweet* of a bird. See also *Woofer.*

Twisted Pair: The name given to a pair of *twisted* copper wires used for telephony. The gauge (and, consequently, the frequency response) of this type of transmission line will depend on the precise purpose and location. The “twist” is to improve common mode noise rejection.

Two’s Complement: The type of arithmetic used by most DSP processors which allows a very convenient way of representing negative numbers, and imposes no overhead on arithmetic operations. In two’s complement the most significant bit is given a negative weighting, e.g.

$$\begin{aligned} 1001\ 0000\ 0000\ 0001_2 &= -2^{15} + 2^{12} + 2^1 \\ &= -32768 + 4096 + 1 = -28671 \end{aligned} \tag{553}$$

See also *Sign bit*.

Two-wire Circuit: A circuit formed of two conductors insulated from each other, providing a send and return path. Signals may pass in one or both directions although not at the same time. See also *Four Wire Circuit, Half Duplex, Full Duplex.*

U

Ungerboeck Codes: See *Trellis Coded Modulation.*

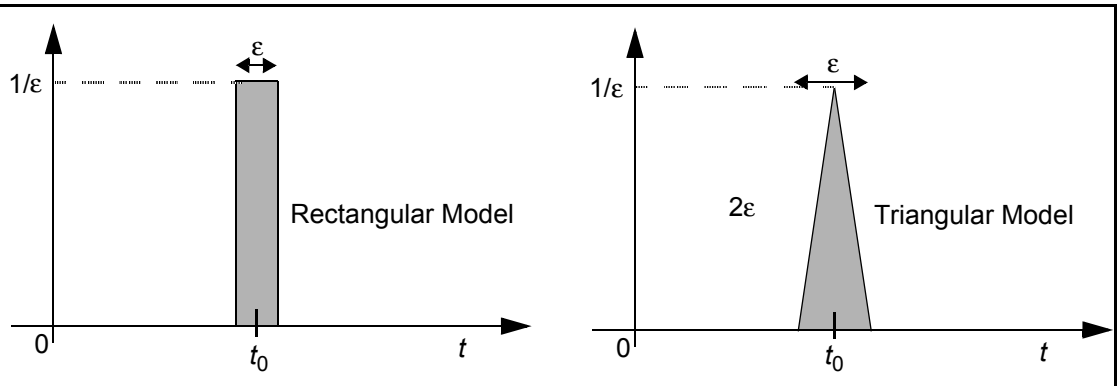
Ultrasonic: Acoustics signals (speed in air, $330\,ms^{-1}$) having frequencies above 20kHz, the upper level of human hearing. The ultrasonic spectrum extends up to MHz frequencies.

Underdetermined System: See *Matrix Properties - Underdetermined System of Equations.*

Unit Impulse Function (Continuous Time and Discrete Time): The mathematical definition of the ***continuous time unit impulse function*** is a signal with an infinite magnitude, but with an infinitesimal duration and that has a unit area. The continuous time unit impulse function is often referred to as the *Dirac impulse* (or *Dirac delta function*) and is not physically realisable. The mathematical representation for the continuous time *unit impulse function* occurring at time t_0, is usually denoted by the Greek letter δ (delta) in the form:

$$\delta(t-t_0) = \begin{cases} 0 & \text{if} \quad t \neq t_0 \\ \text{undefined} & \text{if} \quad t = t_0 \end{cases} \tag{554}$$

Graphically the Dirac impulse, $\delta(t-t_0)$, can be represented as the following rectangular or triangular models where $\varepsilon \to 0$:



Rectangular and triangular models of the *continuous time unit impulse function*. As $\varepsilon \to 0$ both models become infinitely tall and infinitesimally thin, but continue to maintain a unit area.

Although the Dirac impulse does not exist in the real physical world, it does have significant importance in the mathematical analysis of signals and systems. A useful mathematical definition of the continuous time unit impulse function is:

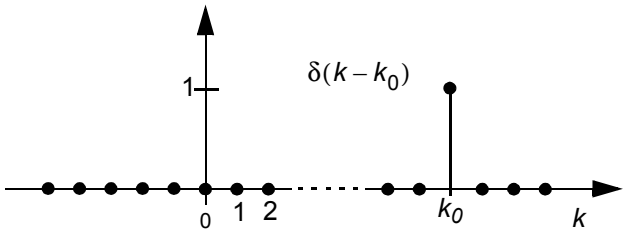
$$\delta(t) = \frac{du(t)}{dt} \tag{555}$$

where *u(t)* is the unit step function. (To be mathematically correct the impulse function is actually a distribution rather than a function of time. The distinction is that a function must be single valued and for any time, *t*, the function has one and only one value.)

The ***discrete time unit impulse function*** has a magnitude of 1 at a specific (discrete) time. The unit impulse response is bounded for all time and is therefore physically realizable. The *discrete time unit impulse function* is often referred to as the *Kronecker impulse* or *(Kronecker delta function).* The mathematical representation for the discrete time *unit impulse function* occurring at (discrete) time k_0, is usually denoted by the Greek letter δ (delta):

$$\delta(k-k_0) = \begin{cases} 0 \text{ if } k \neq k_0 \\ 1 \text{ if } k = k_0 \end{cases} \quad \text{discrete time} \tag{556}$$

Graphically the *discrete time unit impulse function*, $\delta(k-k_0)$, can be represented as:



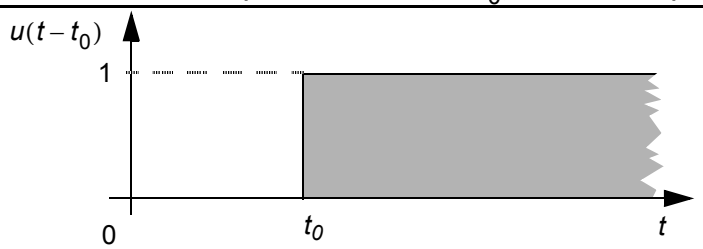
The *discrete time unit impulse function* $\delta(k-k_0)$

Both the discrete time and continuous time *unit impulse functions* exhibit a sampling property when an analog signal is multiplied by a unit impulse response and integrated over time. Hence they are extremely useful mathematical tools for the analysis and definition of DSP sampled data systems. See also *Elementary Signals, Fourier Transform Properties, Impulse, Rectangular Pulse, Sampling Property, Unit Step Function*.

Unit Step Function (Continuous Time and Discrete Time): The mathematical representation for the ***continuous time unit step function*** occurring at time t_0, is usually denoted by the letter *u*, and defined by:

$$u(t-t_0) = \begin{cases} 0 \text{ if } t < t_0 \\ 1 \text{ if } t \geq t_0 \end{cases} \quad \text{continuous time} \tag{557}$$

Graphically the *continuous time unit step function*, $u(t-t_0)$, can be represented as:



The *continuous time unit step function* $u(t-t_0)$

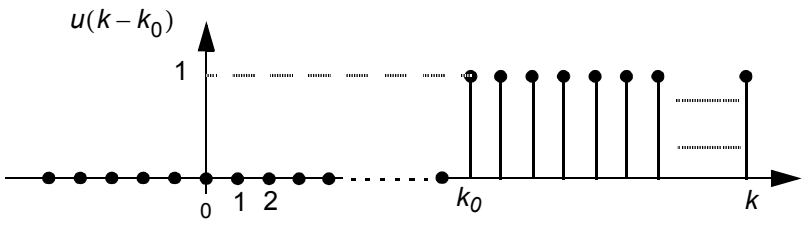
The unit step function can be mathematically derived from the unit impulse function, $\delta(t)$, as:

$$u(t) = \int_{\infty}^{t} \delta(\tau) d\tau \tag{558}$$

The ***discrete time unit step function*** is denoted by:

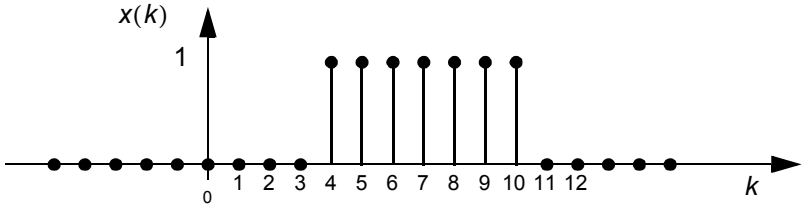
$$u(k-k_0) = \begin{cases} 0 \text{ if } k < 0 \\ 1 \text{ if } k \geq 0 \end{cases} \quad \text{discrete time} \tag{559}$$

Graphically the *discrete time unit step function*, $u(k-k_0)$, can be represented as:



The *discrete time unit step function* $u(k-k_0)$

Rectangular, or pulse functions can be generated by the addition of unit step functions:



$x(k) = u(k-4) - u(k-10)$

See also *Elementary Signals, Fourier Transform Properties, Impulse, Rectangular Pulse, Sampling Property, Step Response, Unit Impulse Function*.

Unit Step Response: See *Step Response*.

Unit Pulse Function: See *Rectangular Pulse, Unit Step Pulse*.

Unit Vector: See *Vector Properties and Definitions - Unit Vector.*

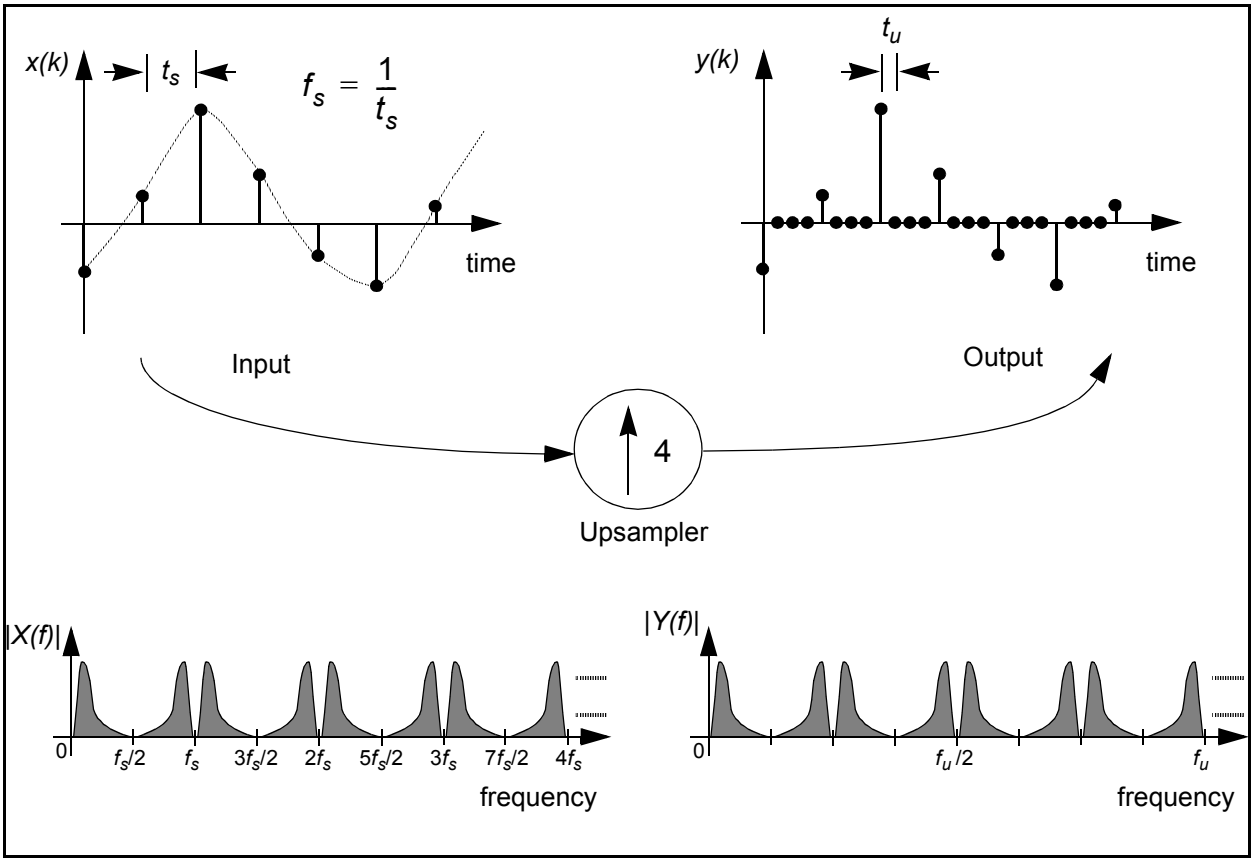
Unitary Matrix: See *Matrix Properties - Unitary.*

Unstable: See *Instability*.

Upper Triangular Matrix: See *Matrix Structured - Upper Triangular.*

Upsampling: Increasing the sampling rate of a digital signal by inserting zeroes between adjacent samples. To upsample a digital signal, x_k, sampled at f_s Hz to Mf_s Hz would require that *M-1* zeroes are inserted between adjacent samples in the original signal. Upsampling in combination with a low pass filter to remove the aliased portions of the frequency spectra gives interpolation. Up-sampling has no effect on the shape of the frequency spectrum of the signal. (If up sampling was performed

using a digital zero order hold, i.e. the value of x_k is inserted instead of zeroes, then the frequency spectrum of the output signal is modulated by a sinc function.) See also *Downsampling, Decimation*, *Fractional Sampling Rate Converter, Interpolation, Sigma Delta Converter, Zero Order Hold.*



V

V-Series Recommendations: The V-series recommendations from the International Telecommunication (ITU), advisory committee on telecommunications (denoted ITU-T, and formerly known as CCITT) propose a number of standards for telecommunication based data transmission. Among the more well known of these standards from a DSP perspective are V22bis (2400 bits/s modem), V32bis (14400 bits/sec modem), V34 (14400 bit/s modem), and V42bis (higher than 28800 bits/s modem featuring data compression) which all feature advanced adaptive signal processing techniques for echo control and data equalisation. Some of the current ITU-T *V-series recommendations* (http://www.itu.ch) can be summarised as:

V.1	Equivalence between binary notation symbols and the significant conditions of a two-condition code.
V.2	Power levels for data transmission over telephone lines.
V.4	General structure of signals of International Alphabet No. 5 code for character oriented data transmission over public telephone networks.
V.7	Definitions of terms concerning data communication over the telephone network.
V.8	Procedures for starting sessions of data transmission over the general switched telephone network.
V.10	Electrical characteristics for unbalanced double-current interchange circuits operating at data signalling rates nominally up to 100 kbit/s.
V.11	Electrical characteristics for balanced double-current interchange circuits operating at data signalling rates up to 10 Mbit/s.
V.13	Simulated carrier control.
V.14	Transmission of start-stop characters over synchronous bearer channels.
V.15	Use of acoustic coupling for data transmission.
V.16	Medical analogue data transmission modems.
V.17	A 2-wire modem for facsimile applications with rates up to 14 400 bit/s.
V.17	A 2-wire modem for facsimile applications with rates up to 14 400 bit/s.
V.18	Operational and interworking requirements for modems operating in the text telephone mode.
V.19	Modems for parallel data transmission using telephone signalling frequencies.
V.21	300 bits per second duplex modem standardized for use in the general switched telephone network.
V.22	1200 bits per second duplex modem standardized for use in the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits.
V.22bis	2400 bits per second duplex modem using the frequency division technique standardized for use on the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits.
V.23	600/1200-baud modem standardized for use in the general switched telephone network.
V.24	List of definitions for interchange circuits between terminal equipment (DTE) and data circuit-terminating equipment (DCE). The V24 standard is very similar to the RS232 standard.
V.25	Automatic answering equipment and/or parallel automatic calling equipment on the general switched telephone network including procedures for disabling of echo control devices for both manual and automatic operation.
V.25bis	Automatic calling and/or answering equipment on the general switched telephone network (GSTN) using the 100-series interchange circuits.
V.26	2400 bits per second modem standardized for use on 4-wire leased telephone-type circuits.
V.26bis	2400/1200 bits per second modem standardized for use in the general switched telephone network.
V.26ter	2400 bits per second duplex modem using the echo cancellation technique standardized for use on the general switched telephone network and on point-to-point 2-wire leased telephone-type circuits.
V.27	4800 bits per second modem with manual equalizer standardized for use on leased telephone-type circuits.

V.27bis	4800/2400 bits per second modem with automatic equalizer standardized for use on leased telephone-type circuits.
V.27ter	4800/2400 bits per second modem standardized for use in the general switched telephone network.
V.28	Electrical characteristics for unbalanced doubled-current interchange circuits.
V.29	9600 bits per second modem standardized for use on point-to-point 4-wire leased telephone-type circuits.
V.31	Electrical characteristics for single-current interchange circuits controlled by contact closure.
V.31bis	Electrical characteristics for single-current interchange circuits using optocouplers.
V.32	A family of 2-wire, duplex modems operating at data signalling rates of up to 9600 bit/s for use on the general switched telephone network and on leased telephone-type circuits.
V.32bis	A duplex modem operating at data signalling rates of up to 14400 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuit.
V.33	14400 bits per second modem standardized for use on point-to-point 4-wire leased telephone-type circuits.
V.34	A modem operating at data signalling rates of up to 28800 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits.
V.36	Modems for synchronous data transmission using 60-108 kHz group band circuits.
V.37	Synchronous data transmission at a data signalling rate higher than 72 kbit/s using 60-108 kHz group band circuits.
V.38	A 48/56/64 kbit/s data circuit terminating equipment standardized for use on digital point-to-point leased circuits.
V.41	Code-independent error-control system.
V.42	Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion.
V.42bis	Data compression procedures for data circuit terminating equipment (DCE) using error correction procedures.
V.50	Standard limits for transmission quality of data transmission.
V.51	Organization of the maintenance of international telephone-type circuits used for data transmission.
V.52	Characteristics of distortion and error-rate measuring apparatus for data transmission.
V.53	Limits for the maintenance of telephone-type circuits used for data transmission.
V.54	Loop test devices for modems.
V.55	Specification for an impulsive noise measuring instrument for telephone-type circuits.
V.56	Comparative tests of modems for use over telephone-type circuits.
V.57	Comprehensive data test set for high data signalling rates.
V.58	Management information model for V-series DCE's.
V.100	Interconnection between public data networks (PDNs) and the public switched telephone networks (PSTN).
V.110	Support of data terminal equipments with V-Series type interfaces by an integrated services digital network.
V.120	Support by an ISDN of data terminal equipment with V-series type interfaces with provision for statistical multiplexing.
V.230	General data communications interface layer 1 specification.

For additional detail consult the appropriate standard document or contact the ITU. See also *Bell 103/113., Bell 202, Bell 212, International Telecommunication Union, ITU-T Modem, Recommendations, Standards*.

Variable Step Size LMS: See *Least Mean Squares Algorithm Variants.*

Variance: The variance of a signal is the mean of the square of the signal about the mean value. If the signal is ergodic the statistical averages will equal the time averages and then:

$$m = \text{Mean of } x(k) = E\{x(k)\} = \sum_k x(k)p\{x(k)\} \cong \frac{1}{N}\sum_{k=0}^{N-1} x(k) \tag{560}$$

and

$$\begin{aligned}\text{Variance of } x(k) &= E\{(x(k)-m)^2\} = \sum_k (x(k)-m)^2 p\{x(k)\} \\ &\cong \frac{1}{N}\sum_{k=0}^{N-1} [x(k)-m]^2\end{aligned} \tag{561}$$

for large N. In a practical DSP situation where real signals are being used, the *variance* is often calculated using time averages. Variance gives a measure of the AC power in a signal. See also *Ergodic, Expected Value, Mean Value, Mean Squared Value, Wide Sense Stationarity.*

Vector: A *vector* is a set of ordered information. A vector is *usually* denoted in texts using boldface lower case letters, $\boldsymbol{v}$ (cf. matrices, denoted ;kby upper case boldface) or with an underscore, $\underline{v}$. A column *vector* has n rows and one column i.e. $n \times 1$ dimension, and a row *vector* has one row and n columns, i.e. $1 \times n$ dimension.

In DSP a vector is usually a set of ordered elements conveying information or data. For example the last N samples of a signal, $\boldsymbol{g}_k$ may be stored in a continuous array of memory and referred to and operated on as a (data) vector:

$$\boldsymbol{g}_k = \begin{bmatrix} g_k \\ g_{k-1} \\ g_{k-2} \\ \vdots \\ g_{k-N+2} \\ g_{k-N+1} \end{bmatrix} \tag{562}$$

Vectors can be added subtracted, multiplied, scaled and transposed. See also *Data Vector, Matrix, Vector Operations, Vector Properties, Weight Vector*.

Vector Addition: See *Vector Operations - Addition.*

Vector-Matrix Multiplication: See *Vector Operations - Matrix-Vector Multiplication.*

Vector Multiplication: See *Vector Operations - Multiplication.*

Vector Operations: Vectors of the appropriate dimension can be added, subtracted, multiplied, scaled, and transposed.

- **Addition (Subtraction):** If two vectors are to be added (or subtracted) then they must be of exactly the same dimension. Each element in one vector is added (subtracted) to the analogous element in the other vector. For example:

$$\begin{bmatrix}1\\6\\2\end{bmatrix}+\begin{bmatrix}3\\0\\3\end{bmatrix}=\begin{bmatrix}(1+3)\\(6+0)\\(2+3)\end{bmatrix}=\begin{bmatrix}4\\6\\5\end{bmatrix} \tag{563}$$

Vector addition is commutative, i.e. $\boldsymbol{a}+\boldsymbol{b}=\boldsymbol{b}+\boldsymbol{a}$.

- **Dot Product:** See *Vector Operations - Inner Product.*
- **Inner Product:** When a row vector is multiplied by a column vector of the same dimension, the result is a scalar called the *inner product*. For example an FIR filter forms an inner product by multiplying the weight vector by the data vector. The inner product is sometimes referred to as the dot product. See also *Outer Product.*

$$\boldsymbol{w}^T\boldsymbol{x}=\begin{bmatrix}w_0 & w_1 & w_2 & w_3\end{bmatrix}\begin{bmatrix}x_k\\x_{k-1}\\x_{k-2}\\x_{k-3}\end{bmatrix} \tag{564}$$

- **Multiplication:** Two vectors, $\boldsymbol{w}$ and $\boldsymbol{v}$, can be multiplied either to form the inner product, $\boldsymbol{w}^T\boldsymbol{v}$ or the outer product, $\boldsymbol{w}\boldsymbol{v}^T$.

  The inner product (also known as the dot product) of an $1\times n$ and an $n\times 1$ vector is a scalar. For example:

$$\begin{bmatrix}w_0 & w_1 & w_2\end{bmatrix}\begin{bmatrix}x_0\\x_1\\x_2\end{bmatrix}=w_0x_0+w_1x_1+w_2x_2 \tag{565}$$

  The outer product of an $n\times 1$ and a $1\times n$ vector is a square $n\times n$ matrix. For example

$$\begin{bmatrix}w_0\\w_1\\w_2\end{bmatrix}\begin{bmatrix}v_0 & v_1 & v_2\end{bmatrix}=\begin{bmatrix}v_0w_0 & v_1w_0 & v_2w_0\\v_0w_1 & v_1w_1 & v_2w_1\\v_0w_2 & v_1w_2 & v_2w_2\end{bmatrix} \tag{566}$$

  The inner product (also known as the dot product) is widely used for digital filter presentation, and the output product is found in a number of linear algebraic derived DSP algorithms such as Recursive Least Squares.

  Matrix-Vector Multiplication: A $n\times 1$ vector can be premultiplied by an $m\times n$ matrix to give an $m\times 1$. vector. For example:

$$\begin{bmatrix}a_{11} & a_{12}\\a_{21} & a_{22}\\a_{31} & a_{32}\end{bmatrix}\begin{bmatrix}b_1\\b_2\end{bmatrix}=\begin{bmatrix}a_{11}b_1+a_2b_2\\a_{21}b_1+a_{22}b_2\\a_{31}b_1+a_{32}b_2\end{bmatrix} \tag{567}$$

- A $1\times n$ vector can be postmultiplied by a $n\times m$ matrix to give a $1\times m$ vector.

$$\begin{bmatrix}b_1 & b_2\end{bmatrix}\begin{bmatrix}a_{11} & a_{21} & a_{31}\\a_{12} & a_{22} & a_{32}\end{bmatrix}=\begin{bmatrix}a_{11}b_1+a_2b_2 & a_{21}b_1+a_{22}b_2 & a_{31}b_1+a_{32}b_2\end{bmatrix} \tag{568}$$

Note that if $\boldsymbol{Ab} = \boldsymbol{c}$ then $\boldsymbol{b}^T\boldsymbol{A}^T = \boldsymbol{c}^T$. See also *Matrix Operations.*

- **Scaling:** A vector, $\boldsymbol{a}$, is scaled by multiplying each element by a scale factor, s.

$$s\boldsymbol{a} = s\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} sa_1 \\ sa_2 \\ sa_3 \end{bmatrix} \tag{569}$$

- **Transpose:** The transpose of a row vector is obtained by writing the top to bottom elements as the left to right elements of a column vector, and vice-versa for the transpose of a column vector. The transpose of a vector $\boldsymbol{a}$ is denoted as $\boldsymbol{a}^{\mathrm{T}}$. For example, if:

$$\boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \qquad \Rightarrow \qquad \boldsymbol{b}^T = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \end{bmatrix} \tag{570}$$

Note that $(\boldsymbol{b}^T)^T = \boldsymbol{b}$.

- **Subtraction:** *Vector Operations - Addition.*
- **Vector-Matrix Multiplication:** See *Vector Operations - Matrix-Vector Multiplication.*

See also *Matrix, Vector Properties and Definitions.*

Vector Properties and Definitions: A number of vector properties can be defined:

- **Basis:** A basis is a minimal set of linearly independent vectors which spans a particular subspace. Representations of any vector in that subspace spanned by the basis vectors can be achieved by a unique linear combination of the basis vectors.
- **Cauchy-Schwartz Inequality:** The Cauchy Schwartz inequality as applies to the 2-norm of two vectors is given by:

$$\|\boldsymbol{w}^T\boldsymbol{x}\| \le \|\boldsymbol{w}\|_2\|\boldsymbol{x}\|_2 \tag{571}$$

A useful interpretation of this inequality is that the output of an FIR digital filter will have a magnitude less than or equal to the multiplication of the 2-norm of the weight vector and data vector; this information can be useful in deciding the wordlength required be a DSP processor. See also *Vector Properties - Norm, FIR Filter.*

- ∞ **-norm:** See *Matrix Properties - Norm.*
- **Linearly Dependent:** See *Linearly Independent Entry.*
- **Linearly Independent:** A set of vectors, $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, is linearly independent if:

$$\sum_{j=1}^{N} \alpha_j \boldsymbol{x}_j = 0 \tag{572}$$

implies that $\alpha_i = 0$, for i = 1 to N If this condition is not true, then the vector set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ is said to be linearly dependent.

As an example consider the vector set:

$$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\} = \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix} \tag{573}$$

There is clearly no linear combination of $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ such that

$$\sum_{j=1}^{3} \alpha_j \boldsymbol{x}_j = \alpha_1 \boldsymbol{x}_1 + \alpha_2 \boldsymbol{x}_2 + \alpha_3 \boldsymbol{x}_3 \neq 0 \tag{574}$$

other than the trivial solution of $\alpha_1 = \alpha_2 = \alpha_3 = 0$. The set of vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ are therefore *linearly independent*. However the set of vectors:

$$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3\} = \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}1\\2\\0\end{bmatrix} \tag{575}$$

are **not** linearly independent (and therefore linear dependent) as:

$$\sum_{j=1}^{3} \alpha_j \boldsymbol{w}_j = \alpha_1 \boldsymbol{w}_1 + \alpha_2 \boldsymbol{w}_2 + \alpha_3 \boldsymbol{w}_3 = 0 \tag{576}$$

if $\alpha_1 = 1, \alpha_2 = 2, \alpha_3 = -1$. See also *Vector Properties - Basis, Subspace, Rank.*

- **Minimum Norm:** A system of linear equations can be defined as:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{577}$$

where $\boldsymbol{A}$ is a known $m \times n$ matrix and has $\text{rank}(\boldsymbol{A}) = \min(m, n)$, $\boldsymbol{x}$ is an unknown n element vector, and $\boldsymbol{b}$ is a known m element vector. If multiple solutions exist that give the same error between $\boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{b}$, then the solution with the minimum 2-norm is typically desirable. This solution is referred to as the *minimum norm solution* and is given by:

$$\boldsymbol{x}_{LS} = \boldsymbol{A}^{+}\boldsymbol{b} \tag{578}$$

where $\boldsymbol{A}^{+}$ is the pseudoinverse. See also *Matrix Properties - Underdetermined/Overdetermined, Pseudo-Inverse, Vector Properties - Norm.*

- **Norm:** The vector norm provides a measure of the magnitude or distance spanned by an n element vector in n-dimensional space. The most useful class of norms are the *p-norms* defined by:

$$\|\boldsymbol{v}\|_p = (|v_1| + |v_2| + \ldots + |v_n|)^{1/p} = \left[\sum_{i=1}^{n} |v_i|\right]^{\frac{1}{p}} \tag{579}$$

The most often used of these norms is the *2-norm*, also referred to as the magnitude of the vector $\boldsymbol{v}$:

$$\|\boldsymbol{v}\|_2 = (v_1^2 + v_2^2 + \ldots + v_n^2)^{1/2} = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2} \tag{580}$$

The square of the 2-norm is denoted as $\|\boldsymbol{v}\|_2^2$.

For example, the 2-norm of a vector, $\boldsymbol{x}$:

$$\boldsymbol{x} = \begin{bmatrix} 3 \\ 4 \\ -7 \end{bmatrix} \qquad \|\boldsymbol{x}\|_2 = \sqrt{(9+16+49)} = \sqrt{74} = 8.602 \tag{581}$$

Other norms occasionally used are the 1-norm, which is the sum of the magnitude of all of the elements, and the and the ∞-norm, which returns the magnitude of the element with the largest absolute value:

$$\|\boldsymbol{v}\|_1 = |v_1| + |v_2| + \ldots + |v_n| \tag{582}$$

$$\|\boldsymbol{v}\|_\infty = \max|x_i| \quad \text{for} \quad i = 1 \text{ to } n"" \tag{583}$$

For the above 3 element vector x, $\|\boldsymbol{x}\|_1 = 14$ and, $\|\boldsymbol{x}\|_\infty = 7$.

A p-norm unit vector is one that $\|\boldsymbol{x}\|_p = 1$ .. See also *Matrix Properties - Invariant Norm.*

- **One Norm:** See *Vector Properties - Norm.*
- **Orthogonal:** A set of vectors $(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \ldots, \boldsymbol{v}_n)$ is said to be orthogonal if:

$$\boldsymbol{v}_i^T \boldsymbol{v}_j = 0 \quad \text{for all } i \neq j \tag{584}$$

- **Orthonormal:** A set of vectors $(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \ldots, \boldsymbol{v}_n)$ is said to be orthonormal if:

$$\boldsymbol{v}_i^T \boldsymbol{v}_j = \delta_{ij} \quad \text{for all } i, j \tag{585}$$

where δ_{ij} is the Kronecker delta (i.e., δ_{ij}=1 if i=j and δ_{ij}=0 otherwise). Orthogonal and orthonormal sets of vectors seem closely related and they are. The important distinction between an orthogonal set of vectors and an orthonormal set of vectors is that the vectors from the orthonormal set all have a norm of one. This is not necessarily the case for the set of orthogonal vectors.

- **Outer Product:** When a column vector $(n \times 1)$ is post-multiplied by a row vector $(1 \times n)$ the result is a matrix ($n \times n$ elements). For example for n = 3:

$$\boldsymbol{x}\boldsymbol{x}^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} = \begin{bmatrix} (x_1^2) & (x_1 x_2) & (x_1 x_3) \\ (x_2 x_1) & (x_2^2) & (x_2 x_3) \\ (x_3 x_1) & (x_3 x_2) & (x_3^2) \end{bmatrix} \tag{586}$$

The outer product is used to realise estimates of the covariance matrix and/or correlation matrix and is widely used in adaptive digital signal processing formulations. See also *Vector Properties - Inner product.*

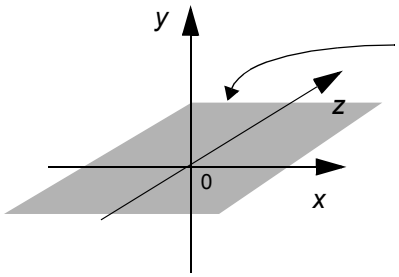
- **Subspace:** Given an m-dimensional space, $\Re^m$, and a set of m-dimensioned vectors $(\boldsymbol{v}_1 \ \boldsymbol{v}_2 \ \boldsymbol{v}_3 \ \ldots \ \boldsymbol{v}_n)$, the set of all possible linear combinations of these vectors forms a *subspace* of $\Re^m$. The form of the linearly combination is given by:

$$\sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i \text{ where, } \alpha_i \in \Re \tag{587}$$

The *subspace* defined by the linear combination of the vectors is said to be the span of $(\boldsymbol{v}_1 \ \boldsymbol{v}_2 \ \boldsymbol{v}_3 \ \ldots \ \boldsymbol{v}_n)$. For example consider the space $\Re^3$. The set of vectors:

$$\boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \tag{588}$$

can only specify points on the *x*-*z* plane within the three dimensional [*x*, *y*, *z*] space. Hence $\boldsymbol{v}_1, \boldsymbol{v}_2$ specify a subspace.

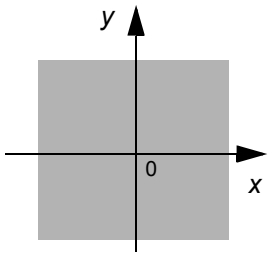


Subspace spanned vectors: $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$

There are effectively an infinite number of (plane) *subspaces* of $\Re^3$. Note that a *subspace* of $\Re^3$ could also be a straight line in three dimensional space if, for example, only $\boldsymbol{v}_1 = [1, 0, 0]^T$ is used to define the *subspace*. Since the form of the linear combination in Eq. [587] allows the scalars to be any value (including all zeros), it is clear that the origin has to be a point in any valid subspace.

- **Space:** Given a vector, $\boldsymbol{v} = [v_1, v_2, \ldots, v_m]^T$, of dimension or length *m*, then it can be said that for $v_i \in \Re$, for $i = 0, 1, 2, \ldots m$ and where $\Re$ is the set of real numbers, then $\boldsymbol{v}$ is contained in the *space* (or *m*-dimensional *space*) denoted as $\Re^m$.

  As examples, the *space* $\Re^2$ can be visualised as the *space* consisting of all points on a two dimensional plane, and the *space* $\Re^3$, considered as all possible points in three dimensional space. For spaces $\Re^4$ and above it is impossible to visualise there physical existence, however their mathematical existence is assured to the reader! See also *Vector Properties - Subspace, Matrix Properties - Range*.



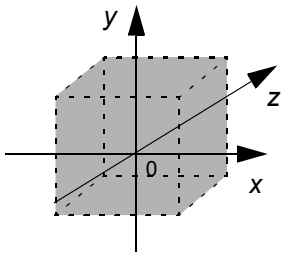
Space $\Re^2$ consists of all points on the x-y plane.

For the vector $\boldsymbol{v} = [x_i \;\; y_i]^T$,

if $x_i, y_i \in \Re$

then $\boldsymbol{v} \in \Re^2$

or $\boldsymbol{v}$ spans the *space* $\Re^2$



Space $\Re^3$ consists of all points in the x-y-z three dimensional space.

For the vector $\boldsymbol{w} = [x_i \;\; y_i \;\; z_i]^T$,

if $x_i, y_i, z_i \in \Re$

then $\boldsymbol{w} \in \Re^3$

or $\boldsymbol{w}$ spans the *space* $\Re^3$

- **Span:** Given a linearly independent set of *m*-dimensional vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n\}$, the set of all linear combinations of these vectors is referred to as the span of $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n\}$, i.e.

$$\text{span}\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n\} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i \;\; \text{where,} \;\; \alpha_i \in \Re \tag{589}$$

Note that the span will define a subspace of $\Re^m$, where $m > n$. Note that if $m = n$ then the vectors span the entire space $\Re^m$. See also *Vector Properties - Space/Subspace*.

- **Transpose Vector:** The transpose of a vector is formed by interchanging the rows and columns and is denoted by the superscript *T*. For example for a vector, $\boldsymbol{x}$:

$$\boldsymbol{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \text{then} \quad \boldsymbol{x}^T = \begin{bmatrix} a & b & c \end{bmatrix} \tag{590}$$

- **2-norm:** See *Vector Properties - Norm*.
- **Unit Vector:** A unit vector with respect to the p-norm is one that $\|\boldsymbol{x}\|_p = 1$. See also *Vector Properties - Norm*.
- **Weight Vector:** The name given to the vector formed by the weights of an FIR filter.

See also *Matrix, Vector Operations.*

Vector Scaling: See *Vector Operations - Scaling.*

Vector Sum Excited Linear Prediction (VSELP): Similar to CELP vocoders except that *VSELP* uses more than one codebook. VSELP also has the additional advantage that it can be run on fixed point DSP processors, unlike CELP which requires floating point computation.

Vector Transpose: See *Vector Operations - Transpose.*

Vibration: A continuous to and fro motion, or reciprocating motion. Vibrations at audible frequencies give rise to sound.

Vibrato: This is a simple frequency modulating effect applied to the output of a musical instrument. For example a mechanical arm on a guitar can be used to frequency modulate the output to produce a warbling effect. Vibrato can also be performed digitally by simple frequency modulation of a signal. See also *Music, Tremolo*.

Virtual Instrument: The terminology used by some companies for a measuring instrument that is implemented on a PC but is presented in a form that resembles the well know analog version of the instrument. For example a virtual oscilloscope forms all of the normal controls as buttons and dials actually drawn on the screen in order that the instrument can immediately be used by an engineer whether they are familiar with DSP or not.

Virtual Reality: A virtual instrument (substitute) for living. Ultimately, this application of DSP image and audio may prove to be very addictive.

Visually Evoked Potential: See *Evoked Potentials*.

Viterbi Algorithm: This algorithm is a means of solving an optimization problem (that can be framed on a trellis -- or structured set of pathways) by calculating the cost (or metric) for each possible path and selecting the path with the minimum metric [103]. The algorithm has proven extremely useful for decoding convolutional codes and trellis coded modulation. For these applications, the paths are defined on a trellis and the metrics are Hamming distance for convolutional codes and Euclidean distance for trellis coded modulation. These metrics result in the smallest possible probability of error when signals are transmitted over an additive white Gaussian noise channel (this is a common modelling assumption in communications). See also *Additive*

White Gaussian Noise (AWGN), Channel Coding, Trellis Coded Modulation, Euclidean Distance, Hamming Distance.

Viterbi Decoder: A technique for decoding convolutionally encoded data streams that uses the Viterbi algorithm (with a Hamming distance metric) to minimize the probability of data errors in a digital receiver. See *Viterbi Algorithm.* See also *Channel Coding.*

VLSI: *Very Large Scale Integration.* The name given to the process of integrating millions of transistors on a single silicon chip to realize various digital devices (logic gates, flip-flops) which in turn are used to make system level components such as microprocessors, all on a single chip.

VME Bus: A bus found in SUN workstations, VAXs and others. Many DSP board manufacturers make boards for *VME bus*, although they are usually a little more expensive than for the PC-Bus.

Vocoders: A *vocoder* analyzes the spectral components of speech to try to identify the parameters of the speech waveform that are perceived by the human ear. These parameters are then extracted, transmitted and used at the receiver to synthesize (approximately) the original speech pattern. The resulting waveform may differ considerably from the original, although it will sound like the original speech signal. Vocoders have become popular at very low bit rates (2.4kbits/sec).

Volatile: Semiconductor Memory that loses its contents when the power is removed is volatile. See also *Non-Volatile, Dynamic RAM, Static RAM.*

Volterra Filter: A filter based on the non linear Volterra series, and used in DSP to model certain types of non-linearity. The second order *Volterra filter* includes second order terms such that the output of the filter is given by:

$$y(k) = \sum_{n=0}^{N-1} w_n(k)x(k-n) + \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} w_{ij}x(k-i)x(k-j) \qquad (591)$$

where w_n are the linear weights and w_{ij} are the quadratic weights. Adaptive LMS based *Volterra filters* are also widely investigated and a good tutorial article can be found in [109].

Voice Grade Channel: A communications channel suitable for transmission of speech, analog data, or facsimile, generally over a frequency band from 300Hz to 3400Hz.

Volume Unit (VU): *VU* meters have been used in recording for many years and give a measure of the relative loudness of a sound [14], [46]. In general a sound of long duration is actually perceived by the human ear as louder than a short duration burst of the same sound. *VU* meters have rather a “sluggish” mechanical response, and therefore have an in built capability to model the human ear temporal loudness response. An ANSI standard exists for the design of *VU* meters. See also *Sound Pressure Level*.

Von Hann Window: See *Windows.*

VXI Bus: A high performance bus used with instruments that can fit on a single PCB card. This standard is a capable of transmitting data at up to 10Mbytes/sec.

W

Waterfall Plot: A graphical 3-D plot that shows frequency plotted on the X-axis, signal power on the Y-axis, and time elapsing on the Z-axis (into the computer screen). As time elapses and segments of data are transformed by the FFT, the screen can appear like a waterfall the 2-D spectra pass along the Z-axis.

Warble Tone: If an audible pure tone is frequency modulated (FM) by a smaller pure tone (typically a few Hz) the perceived signal is often referred to as a *warble tone*, i.e. the signal is perceived to be varying between two frequencies around the carrier tone frequency. *Warble tones* are often used in audiometric testing where stimuli signals are played to a subject through a loudspeaker in a testing room. If pure tones were used there is a possibility that a zone of acoustic destructive interference would occur at or near the patient's head thus making the test erroneous. The use of *warble tones* greatly reduces this possibility as the zones of destructive interference will not be static.

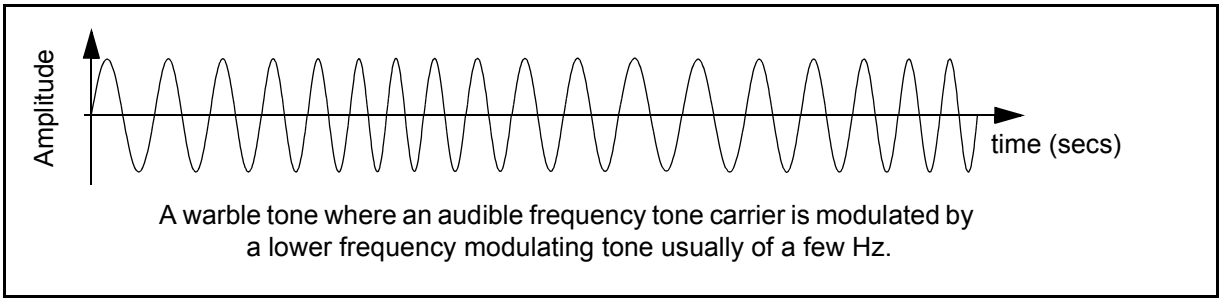
To produce a *warble tone*, consider a carrier tone at frequency f_c, frequency modulated by another tone at frequency f_m:

$$w(t) = \sin(2\pi f_c t + \beta \sin 2\pi f_m t) = \sin\theta(t) \quad \text{i.e. } \theta(t) = 2\pi f_c t + \beta \sin 2\pi f_m t \tag{592}$$

where β is the modulation index which controls the maximum frequency deviation from the carrier frequency. For example if a carrier tone $f_c = 1000$ Hz is to be modulated by a tone $f_m = 5$ Hz such that the *warble tone* signal frequency varies between 900Hz and 1000Hz at a rate 5 times per second, then noting that the instantaneous frequency of an FM tone, f, is given by:

$$f = \frac{1}{2\pi}\frac{d\theta(t)}{dt} = f_c + \beta f_m \cos 2\pi f_m t \tag{593}$$

the modulation index required is $\beta = 20$ to give the required frequency swing. See also *Audiometer, Audiometry, Binaural Beats, Constructive Interference, Destructive Interference*.



A warble tone where an audible frequency tone carrier is modulated by a lower frequency modulating tone usually of a few Hz.

Watt: The surname of the Scottish engineer James *Watt* who gave his name to the unit of power. In an electrical system power is calculated from:

$$P = V \cdot I = I^2 = \frac{V^2}{R} \tag{594}$$

Waveform: The representation of a signal plotted (usually) as voltage against time, where the voltage will represent some analog time varying quantity (e.g. audio, speech and so on).

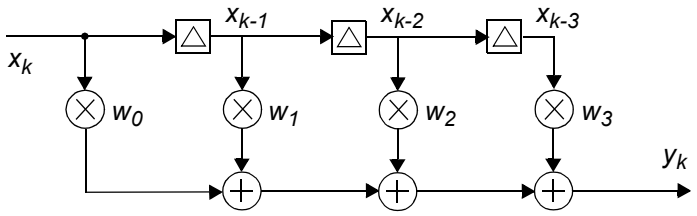
Waveform Averaging: (Ensemble Averaging) The process of taking a number of measurements of a periodic signal, summing the respective elements in each record and dividing by the number of measurements. *Waveform Averaging* is often used to reduce the noise when the noise and periodic signal are uncorrelated. As an example, averaging is widely used in ECG signal analysis where the process retains correlated frequencies in the periodic signal and the removes the uncorrelated one to reveal the distinctive ECG complex.

Wavelet Transform: The *wavelet transform* is an operation that transforms a signal integrated with specific functions, often known as the kernel functions. This kernel functions may be referred to as the mother wavelet and the associated scaling function. Using the scaling function and mother wavelet, multi-scale translations and compressions of these functions can be produced. The *wavelet transform* actually generalizes the time frequency representation of the short time Fourier Transform (STFT). Compared to the STFT the *wavelet transform* allows non-uniform bandwidths or frequency bins and allows resolution to be different at different frequencies. Over the last few years DSP has seen considerable interest and application of the *wavelet transform*, and the interested reader is referred to [49].

Web: See *World Wide Web.*

Weight Vector: Weighted Moving Average (WMA): See *Finite Impulse Response (FIR) filter*. See also *Moving Average.*

Weight Vector: The weights of an FIR digital filter can be expressed in vector notation such that the output of a digital filter can be conveniently expressed as a row-column vector product (or inner product).

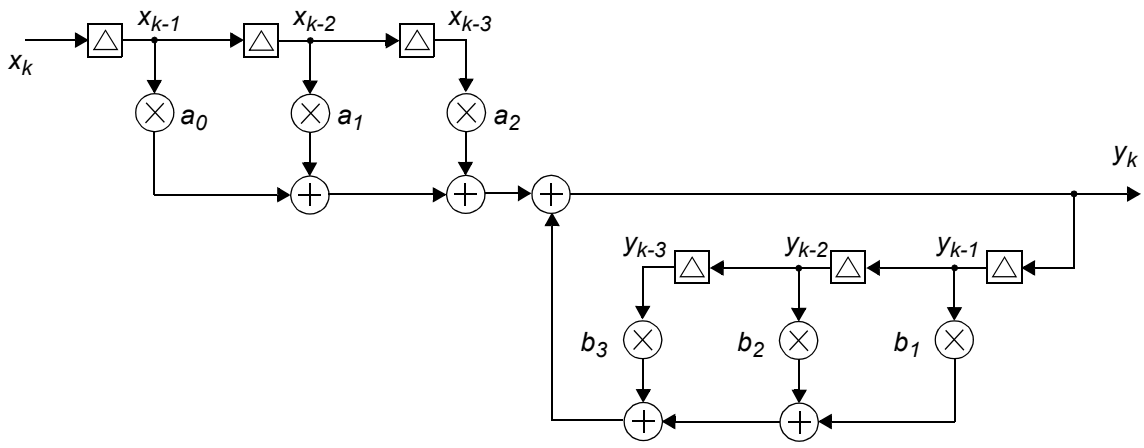


$$y_k = \sum_{n=0}^{3} w_n x_{k-n} = w_0 x_k + w_1 x_{k-1} + w_2 x_{k-2} + w_3 x_{k-3}$$

$$\Rightarrow y_k = \boldsymbol{w}^T \boldsymbol{x} = \begin{bmatrix} w_0 & w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \\ x_{k-2} \\ x_{k-3} \end{bmatrix}$$

If the digital filter is IIR, then two weight vectors can be defined: one for the feedforward weights and one for the feedback weights. For further notational brevity the two weight vectors and two data

vectors can be respectively combined into a single weight vector, and a data vector consisting of past input data and past output samples:.



$$y_k = \sum_{n=0}^{2} a_n x_{k-n} + \sum_{n=1}^{3} b_n y_{k-n} = a_0 x_k + a_1 x_{k-1} + a_2 x_{k-2} + b_1 y_{k-1} + b_2 y_{k-2} + b_3 y_{k-3}$$

$$\Rightarrow y_k = \boldsymbol{a}^T \boldsymbol{x}_k + \boldsymbol{b}^T \boldsymbol{y}_{k-1} = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \\ x_{k-2} \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ y_{k-3} \end{bmatrix}$$

$$\Rightarrow y_k = \begin{bmatrix} \boldsymbol{a}^T & \boldsymbol{b}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{y}_{k-1} \end{bmatrix} = \boldsymbol{w}^T \boldsymbol{u}_k$$

See also *Vector Properties and Definitions - Weight Vector.*

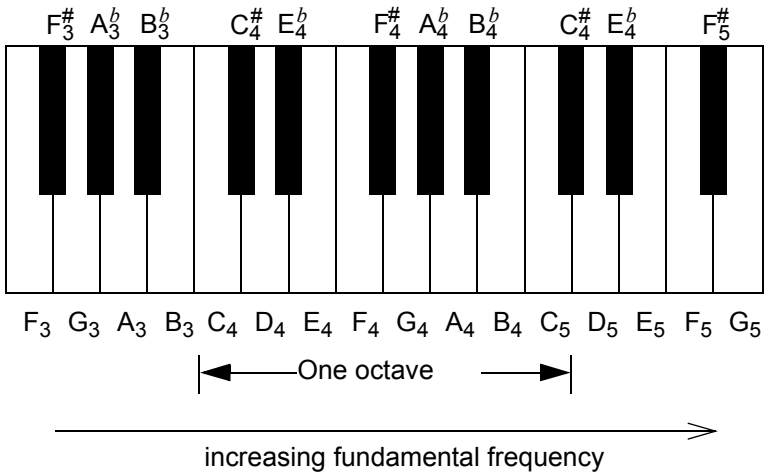
Weighting Curves: See *Sound Pressure Level Weighting Curves*.

Weights: The name given to the multipliers of a digital filter. For example, a particular FIR may be described as having 32 *weights*. The terms *weights* and *coefficients* are used interchangeably. See also *FIR filter, IIR filter, Adaptive Filter.*

Well-Conditioned Matrix: See *Matrix Properties - Well Conditioned.*

Western Music Scale: The *Western music scale* is based around musical notes separated by octaves [14]. If a note, X, is an octave higher than another note, Y, then the fundamental frequency of X is twice that of Y. From one octave frequency to the next in the *Western music scale*, there are twelve equitempered frequencies which are spaced one semitone apart, where a semitone is a logarithmic increase in frequency (If the two octave frequencies are counted then there are thirteen

notes). The *Western music scale* can be best illustrated on the well known piano keyboard which comprises a full chromatic scale:



A section of the familiar piano keyboard with the names of the notes marked. One octave is twelve equitempered notes (sometimes called the chromatic scale), or eight notes of a major scale. The black keys represent various sharps (#) and flats (*b*). The piano keyboard extends in both directions repeating the same twelve note scale. Neighboring keys (black or white) are defined as being a semitone apart. If one note separates two keys, then they are a tone apart. The letters A to G are the names given to the notes.

The International Pitch Standard defines the fundamental frequency of the note A_4 as being 440 Hz. The note A_4 is the first A above middle C (C_4) which is located near the middle of a piano keyboard. Each note on the piano keyboard is characterised by its fundamental frequency, f_0, which is usually the loudest component caused by the fundamental mode of vibration of the piano string being played. The "richness" of the sound of a single note is caused by the existence of other modes of vibration which occur at harmonics (or integer multiples) of the fundamental, i.e. $2f_0$, $3f_0$ and so on. The characteristic sound of a musical instrument is produced by the particular harmonics that make up each note.

On the equitempered *Western music scale* the logarithmic difference between the fundamental frequencies of all notes is equal. Therefore noting that in one octave the frequency of the thirteenth note in sequence is double that of the first note, then if the notes are equitempered the ratio of the fundamental frequencies of adjacent notes must be $2^{1/12} = 1.0594631\ldots$. As defined the ratio between the first and thirteenth note is then of course $(2^{1/12})^{12} = 2$, or an octave. The actual logarithmic difference in frequency between two adjacent notes on the keyboard is:

$$\log 2^{1/12} = 0.025085\ldots \tag{595}$$

Two adjacent notes in the *Western music scale* are defined as being one semitone apart, and two notes separated by two semitones are a tone apart. For example, musical notes B and C are a semitone apart, whereas G and A are a tone apart as they are separated by A^b.

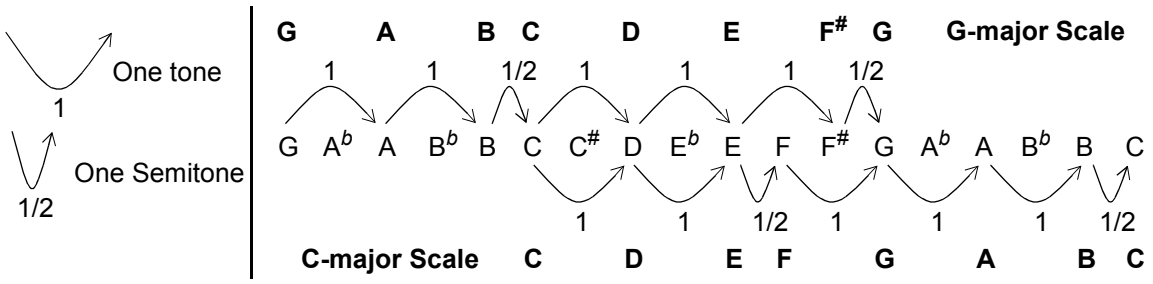
Therefore the fundamental frequencies of 3 octaves of the *Western music scale* can be summarised in the following table, where the fundamental frequency of the next semitone is calculated by multiplying the current note fundamental frequency by 1.0594631...:

Note	Fundamental frequency (Hz)	Note	Fundamental frequency (Hz)	Note	Fundamental frequency (Hz)
C_3	130.812	C_4	261.624	C_5	523.248
$C_3^{\#}$	138.591	$C_4^{\#}$	277.200	$C_5^{\#}$	554.400
D_3	146.832	D_4	293.656	D_5	587.312
E_3^{b}	155.563	E_4^{b}	311.124	E_5^{b}	622.248
E_3	164.814	E_4	329.648	E_5	659.296
F_3	174.614	F_4	349.228	F_5	698.456
$F_3^{\#}$	184.997	$F_4^{\#}$	370.040	$F_5^{\#}$	740.080
G_3	195.998	G_4	392.040	G_5	784.080
A_3^{b}	207.652	A_4^{b}	415.316	A_5^{b}	830.632
A_3	220	A_4	440	A_5	880
B_3^{b}	233.068	B_4^{b}	466.136	B_5^{b}	932.327
B_3	246.928	B_4	493.856	B_5	987.767

A correctly tuned musical instrument will therefore produce notes with the frequencies as stated above. However it is the existence of subtle fundamental frequency harmonics that gives every instrument its unique sound qualities. It is also worth noting that certain instruments may have some or all notes tuned “sharp” or “flat” to create a desired effect. Also noting that pitch perception and frequency is not a linear relationship the high frequencies of certain instruments may be tuned slightly “sharp”.

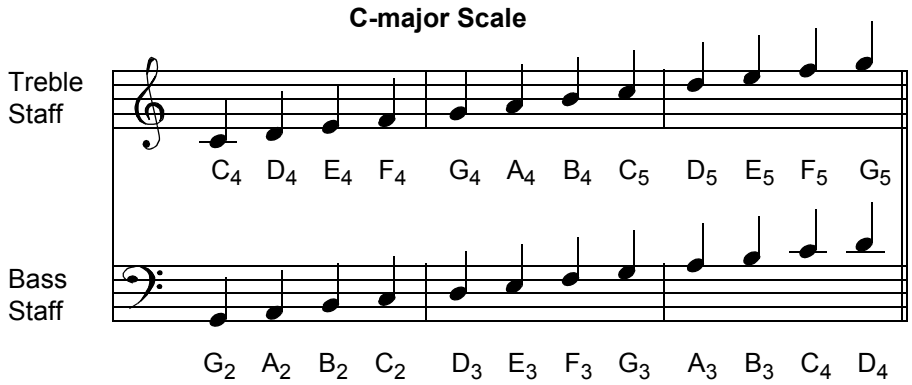
Music is rarely represented in terms of its fundamental frequencies and instead music staffs are used to represent the various notes that make up a particular composition. A piece of music is usually played in a particular musical key which is a subset of eight notes of an octave and where those eight notes have aesthetically pleasing perceptible qualities. The major key scales are

realised by starting at a root note and selecting the other notes of the key in intervals of 1, 1, 1/2, 1, 1, 1, 1/2 tones (where 1/2 tone is a semitone). For example the C-major and G-major scales are:

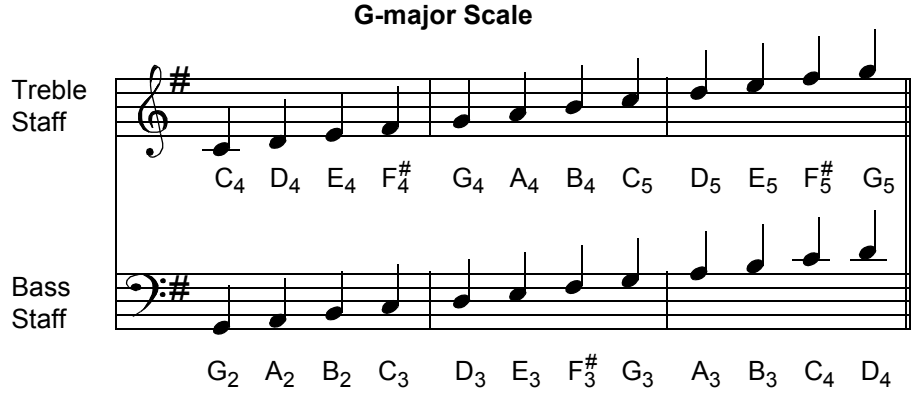

One tone
1
One Semitone
1/2
G A B C D E F# G G-major Scale
1 1 1/2 1 1 1 1/2
G Ab A Bb B C C# D Eb E F F# G Ab A Bb B C
1 1 1/2 1 1 1 1/2
C-major Scale C D E F G A B C

Starting at the any root note, X, of the chromatic scale, the, X-major scale can be produced by selecting notes in steps of 1,1,1/2,1,1,1,1/2 tones. The above shows example of the C- and G-major scales. There are a total of 12 major scales possible.

There are many other forms of musical keys, such as the natural minors which are formed by the root note and then choosing in steps of 1, 1/2, 1, 1, 1/2, 1, 1. For more information on the rather elegant and simple mathematics of musical keys, refer to a text on music theory.



Music notation for the C major scale which has no sharps or flats (i.e., only the white notes of the piano keyboard). Different notes are represented by different lines and spaces on the staff (the five parallel lines). The treble clef (the “g” like letter marking the G-line on the top left hand side of the staff) usually defines the melody of a tune, whereas the bass clef (the “f” like letter marking the F-line on the bottom left hand side of the staff) defines the bass line. Note that middle C (C_4) is represented on a “ledger” line existing between the treble and bass staffs. On a piano the treble is played with the right hand, and the bass with the left hand. For other scales (major or minor), the required sharps and flats are shown next to the bass and treble clefs. Many musical instruments only have the capability of playing either the treble or bass, e.g. the flute can only play the treble clef, or the double bass can only play the bass clef.



Music notation for the G major scale which has one sharp (sharps and flats are the black notes of the piano keyboard). Therefore whenever an F note is indicated by the music, then an $F^{\#}$ should be played in order to ensure that the G-major scale is used.

So what are the qualities of the *Western music scale* that make it pleasurable to listen to? The first reason is familiarity. We are exposed to music from a very early age and most people can recognise and recall a simple major scale or a tune composed of notes from a major scale. The other reasons are that the ratios of the frequencies of certain notes when played together are "almost" low integer ratios and these chords of more than one note take on a very "full" sound.

For example the C-major chord is composed of the 1st, 3rd and 5th notes of the C-major scale, i.e. C,E,G. If we consider the ratios of the fundamental frequencies of these notes:

$$\begin{aligned} \frac{C}{E} &= 2^{4/12} = 1.2599\ldots \approx \frac{5}{4} \\ \frac{C}{G} &= 2^{7/12} = 1.4983\ldots \approx \frac{3}{2} \\ \frac{E}{G} &= 2^{3/12} = 1.189\ldots \approx \frac{6}{5} \end{aligned} \tag{596}$$

they can be approximated by "almost" integer ratios of the fundamental frequencies. (Note that on the very old scales -- the Just scale and the Pythagorean scale -- these ratios were exact). When these three notes are played together the frequency differences actually reinforce the fundamental which produces a rich strong sound. This can be seen by considering the simple trigonometric identities:

$$\begin{aligned} C + E &= \cos C_0 + \cos(2^{1/3} C_0) \approx 2\cos\left(\frac{1-5/4}{2}\right) C_0 \cos\left(\frac{1+5/4}{2}\right) C_0 \\ &= 2\cos\frac{1}{8} C_o \cos\frac{9}{8} C_0 \end{aligned} \tag{597}$$

and

$$\begin{aligned} C+G &= \cos C_0 + \cos(2^{7/12}C_0) \approx 2\cos\left(\frac{1-3/2}{2}\right)C_0\cos\left(\frac{1+3/2}{2}\right)C_0 \\ &= 2\cos\frac{1}{4}C_0\cos\frac{5}{4}C_0 \end{aligned} \tag{598}$$

where $C_0 = 2\pi f_C t$ and f_C is the fundamental frequency of the C note. Adding together the C and E results in a sound that may be interpreted as a C three octaves below C_0 modulating a D. Similarly the addition of the C and G results in sound that may be interpreted as a C two octaves below C_0 that is modulated an E. The existence of these various modulating subharmonics leads to the "full" and aesthetically pleasing sound of the chord. In addition to major chords, there are many others such as the minor, the seventh and so on. All of the chords have there own distinctive sound to which we have become accustomed and associated certain styles of music.

Prior to the existence of the equitempered scale there were other scales which used perfect integer ratios between notes ratios. Also around the world there are still many other music scales to be found, particularly in Asia. See also *Digital Audio, Just Music Scale, Music, Music Synthesis, Pythagorean Scale*.

White Noise: A signal that (in theory) contains all frequencies and is (for most purposes) completely unpredictable. Most white noise is defined as being Gaussian, which means that it has definable properties of mean (average value) and variance (a measure of its power). White noise has a constant power per unit bandwidth, and is labelled *white* because of the analogy with white light (containing all visible light frequencies with nearly equal power). In a digital system, a *white noise* sequence has a flat spectrum from 0Hz to half the sampling frequency.

Wide Sense Stationarity: If a discrete time signal, $x(k)$, has a time invariant mean:

$$E\{x(k)\} = \sum_k x(k)p\{x(k)\} \tag{599}$$

and a time invariant autocorrelation function:

$$r(n) = \sum_k x(k)x(k-n)p\{x(k)\} \tag{600}$$

that is a function only of the time separation, $n-k$, but not of time, k, is said to be *wide sense stationary*. Therefore if the signal, $x(k)$, is also ergodic, then:

$$E\{x(k)\} \cong \frac{1}{M_2 - M_1}\sum_{n=M_1}^{M_2-1} x(k), \quad \text{for any } M_1 \text{ and } M_2 \text{ where } M_2 \gg M_1 \tag{601}$$

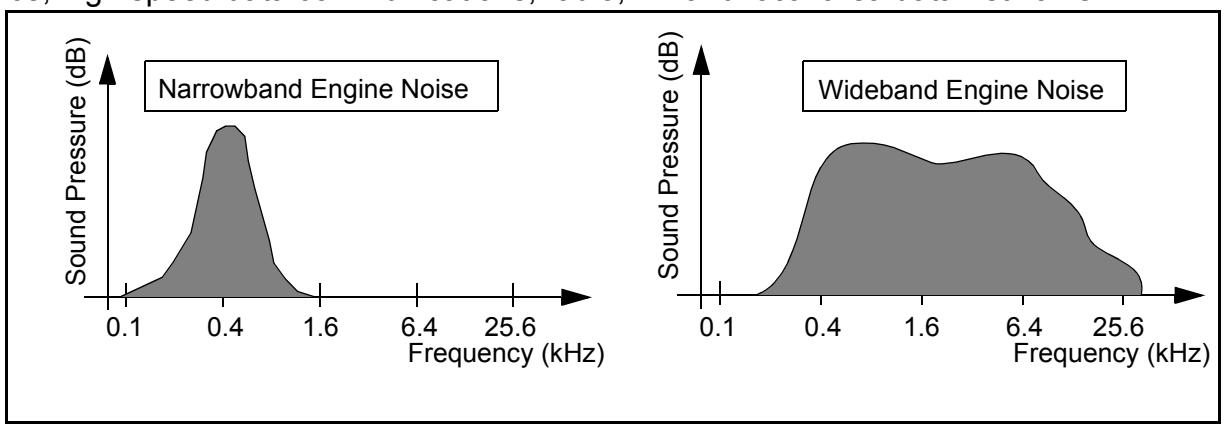
and

$$E\{x^2(k)\} \cong \frac{1}{M_2 - M_1}\sum_{n=M_1}^{M_2-1} [x(k)]^2, \quad \text{for any } M_1 \text{ and } M_2 \text{ where } M_2 \gg M_1 \tag{602}$$

For derivation and subsequent implementation of least means squares DSP algorithms using stochastic signals, assuming wide sense stationarity is usually satisfactory. See *Autocorrelation, Expected Value, Least Mean Squares, Mean Value, Mean Squared Value, Strict Sense Stationary, Variance, Wiener-Hopf Equations.*

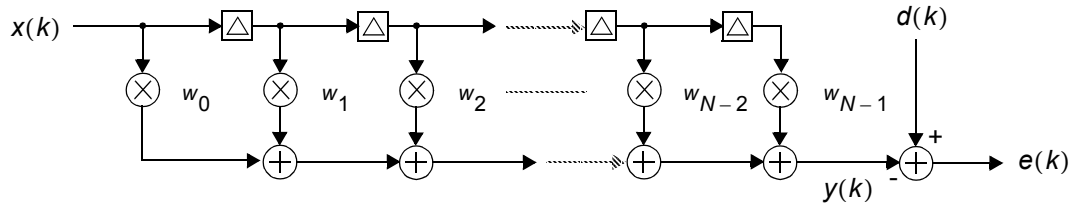
Wideband: A signal that uses a large portion of a particular frequency band may be described as wideband. The classification into *wideband* and narrowband depends on the particular application being described. For example, the noise from a reciprocating (piston) engine may be described as narrowband as it consists of a one main frequency (the drone of the engine) plus a some frequency components around this frequency, whereas the noise from a jet engine could be described as *wideband* as it covers a much larger frequency band and is more white (random) in its make-up.

In telecommunications wideband or broadband may describe a circuit that provides more bandwidth than a voice grade telephone line (300-3000Hz) i.e. a circuit or channel that allows frequencies of upto 20kHz to pass. These type of telecommunication broadband channels are used for voice, high speed data communications, radio, TV and local area data networks.



Widrow: Professor Bernard Widrow of Stanford University, USA, generally credited with developing the LMS algorithm for adaptive digital signal processing systems. The LMS algorithm is occasionally referred to as Widrow's algorithm.

Wiener-Hopf Equations: Consider the following architecture based on a FIR filter and a subtraction element:



The output of an FIR filter, $y(k)$ is subtracted from a desired signal, $d(k)$ to produce an error signal, $e(k)$. If there is some correlation between the input signal, $x(k)$ and the desired signal, $d(k)$ then values can be calculated for the filter weights, $w(0)$ to $w(N-1)$ in order to minimize the mean squared error, $E\{e^2(k)\}$.

If the signal $x(k)$ and $d(k)$ are in some way correlated, then certain applications and systems may require that the digital filter weights, $w(0)$ to $w(N-1)$ are set to values such that the power of the error signal, $e(k)$ is minimised. If weights are found that minimize the error power in the mean squared sense, then this is often referred to as the Wiener-Hopf solution.

To derive the Wiener Hopf solution it is useful to use a vector notation for the input vector and the weight vector. The output of the filter, *y(k)*, is the convolution of the weight vector and the input vector:

$$y(k) = \sum_{n=0}^{N-1} w_n x(k-n) = \boldsymbol{w}^T \boldsymbol{x}(k) \tag{603}$$

where,

$$\boldsymbol{w} = [w_0 \;\; w_1 \;\; w_2 \;\; \ldots \; w_{N-2} \;\; w_{N-1}]^T \tag{604}$$

and,

$$\boldsymbol{x}(k) = [x(k) \;\; x(k-1) \;\; x(k-2) \;\; \ldots \; x(k-N+2) \;\; x(k-N+1)]^T \tag{605}$$

Assuming that $x(k)$ and $d(k)$ are wide sense stationary processes and are correlated in some sense, then the error, $e(k) = d(k) - y(k)$ can be minimised in the mean squared sense.

To derive the Wiener-Hopf equations consider first the squared error:

$$\begin{aligned} e^2(k) &= [d(k) - y(k)]^2 \\ &= d^2(k) - [\boldsymbol{w}^T \boldsymbol{x}(k)]^2 - 2d(k)\boldsymbol{w}^T \boldsymbol{x}(k) \\ &= d^2(k) - \boldsymbol{w}^T \boldsymbol{x}(k)\boldsymbol{x}^T(k)\boldsymbol{w} - 2\boldsymbol{w}^T d(k)\boldsymbol{x}(k) \end{aligned} \tag{606}$$

Taking expected (or mean) values we can write the mean squared error (MSE), $E\{e^2(k)\}$ as:

$$E\{e^2(k)\} = E\{d^2(k)\} - \boldsymbol{w}^T E\{\boldsymbol{x}(k)\boldsymbol{x}^T(k)\}\boldsymbol{w} - 2\boldsymbol{w}^T E\{d(k)\boldsymbol{x}(k)\} \tag{607}$$

Writing in terms of the $N \times N$ correlation matrix,

$$\boldsymbol{R} = E\{\boldsymbol{x}(k)\boldsymbol{x}^T(k)\} = \begin{bmatrix} r_0 & r_1 & r_2 & \ldots & r_{N-1} \\ r_1 & r_0 & r_1 & \ldots & r_{N-2} \\ r_2 & r_1 & r_0 & \ldots & r_{N-3} \\ : & : & : & \ldots & : \\ r_{N-1} & r_{N-2} & r_{N-3} & \ldots & r_o \end{bmatrix} \tag{608}$$

and the $N \times 1$ cross correlation vector,

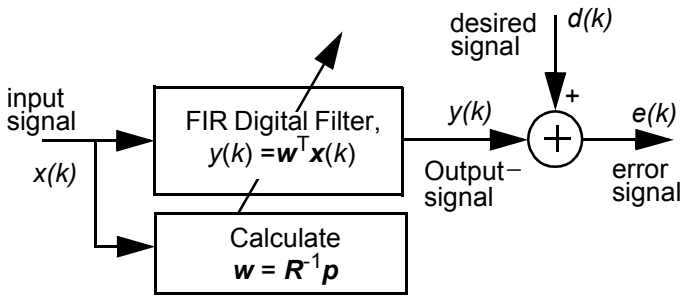
$$\boldsymbol{p} = E\{d(k)\boldsymbol{x}(k)\} = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_{N-1} \end{bmatrix} \tag{609}$$

gives,

$$\zeta = E\{e^2(k)\} = E\{d^2(k)\} + \boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w} - 2\boldsymbol{w}^T\boldsymbol{p} \tag{610}$$

where ζ is used for notational convenience to denote the MSE performance surface. Given that this equation is quadratic in $\boldsymbol{w}$ then there is only one minimum value. The minimum mean squared error (MMSE) solution, $\boldsymbol{w}_{opt}$, can be found by setting the (partial derivative) gradient vector, ∇, to zero:

$$\begin{aligned} \nabla = \frac{\partial \zeta}{\partial \boldsymbol{w}} &= 2\boldsymbol{R}\boldsymbol{w} - 2\boldsymbol{p} = \boldsymbol{0} \\ \Rightarrow \boldsymbol{w}_{opt} &= \boldsymbol{R}^{-1}\boldsymbol{p} \end{aligned} \tag{611}$$



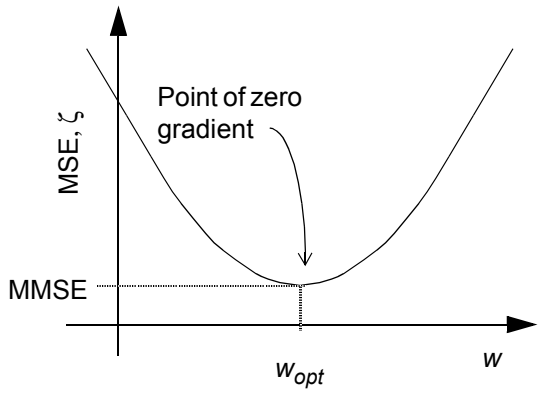
A simple block diagram for the Wiener-Hopf calculation. Note that there is no feedback and therefore, assuming $\boldsymbol{R}$ is non-singular, the algorithm is unconditionally stable.

(612)

To appreciate the quadratic and single minimum nature of the error performance surface consider the trivial case of a one weight filter:

$$\zeta = E\{d^2(k)\} + rw^2 - 2wp \tag{613}$$

where $E[d^2(k)]$, r, and p are all constant scalars. Plotting mean squared error (MSE), ζ, against the weight vector, w, produces a parabola (upfacing):



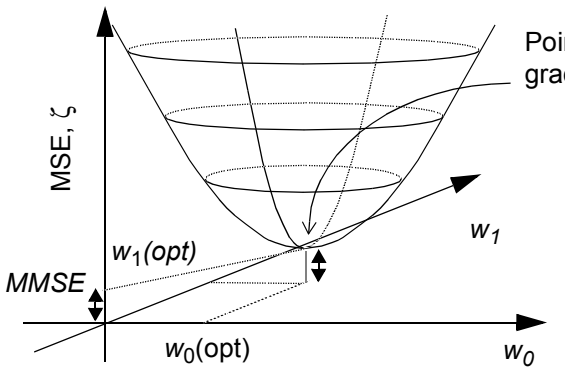
$$\nabla = \frac{d\zeta}{dw} = 2rw - 2p = 0$$

$$w_{opt} = r^{-1}p$$

The mean square error (MSE) performance surface, ζ, of for a single weight filter.

The MMSE solution occurs when the surface has gradient, $\nabla = 0$.

If the filter has two weights the performance surface is a paraboloid which can be drawn in 3 dimensions:



$$\nabla = \frac{d\zeta}{d\boldsymbol{w}} = 2\boldsymbol{R}\boldsymbol{w} - 2\boldsymbol{p} = 0$$

$$\boldsymbol{w}_{opt} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}_{opt} = \begin{bmatrix} r_0 & r_1 \\ r_1 & r_0 \end{bmatrix}^{-1} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix}$$

The mean square error (MSE) performance surface, ζ, of for a two weight filter.

If the filter has more than three weights then we cannot draw the performance surface in three dimensions, however, mathematically there is only one minimum point which occurs when the gradient vector is zero. A performance surface with more than three dimensions is often called a hyperparaboloid.

To actually calculated the Wiener-Hopf solution, $\boldsymbol{w}_{opt} = \boldsymbol{R}^{-1}\boldsymbol{p}$ requires that the $\boldsymbol{R}$ matrix and $\boldsymbol{p}$ vector are realised from the data $x(k)$ and $d(k)$, and the $\boldsymbol{R}$ matrix is then inverted prior to premultiplying vector $\boldsymbol{p}$. Given that we assumed that $x(k)$ and $d(k)$ are stationary and ergodic, then we can estimate all elements of $\boldsymbol{R}$ and $\boldsymbol{p}$ from:

$$r_n = \frac{1}{M}\sum_{i=0}^{M-1} x_i x_{i+n} \qquad \text{and} \qquad p_n = \frac{1}{M}\sum_{i=0}^{M-1} d_i x_{i+n} \tag{614}$$

Calculation of $\boldsymbol{R}$ and $\boldsymbol{p}$ requires approximately $2MN$ multiply and accumulate (MAC) operations where M is the number of samples in a "suitably" representative data sequence, and N is the

adaptive filter length. The inversion of $\boldsymbol{R}$ requires around N^3 MACs, and the matrix-vector multiplication, N^2 MACs. Therefore the total number of computations in performing this one step algorithm is $2MN + N^3 + N^2$ MACs. The computation load is therefore very high and real time operation is computationally expensive. More importantly, if the statistics of signals $x(k)$ or $d(k)$ change, then the filter weights will need to be recalculated, i.e. the algorithm has no tracking capabilities. Hence direct implementation of the Wiener-Hopf solution is not practical for real time DSP implementation because of the high computational load, and the need to recalculate when the signal statistics change. For this reason real time systems which need to minimize an error signal power use gradient descent based adaptive filters such as the least mean squares (LMS) or recursive least squares (RLS) type algorithms. See also *Adaptive Filter, Correlation Matrix, Correlation Vector, Least Mean Squares Algorithm, Least Squares.*

Whitening Filter: A filter that takes a stochastic signal and produces a white noise output [77]. If the input stochastic signal is an autoregressive process, the whitening filters are all-zero FIR filters. See also *Autoregressive Model*.

Window: A *window* is a set of numbers that multiply a set of *N* adjacent data samples. If the data was sampled at frequency f_s, then the window weights N/f_s second of data. There a number of semi-standardized data weighting *windows* used to pre-weight data prior to frequency domain calculations (FFT/DFT). The most common are the Bartlett, Von Hann, Blackman, Blackmann-harris, Hamming, and Hanning:

- **Bartlett Window:** A data weighting window used prior to frequency transformation (FFT) to reduce spectral leakage. Compared to the uniform window (no weighting) the *Bartlett window* doubles the width of the main lobe, while attenuating the main sidelobe by 26dB, compared to the 13dB of the uniform window. For *N* data samples, the Barlett window is defined by:

$$h(n) = 1.0 - \frac{|n|}{N/2} \quad \text{for} \quad n = -\frac{N}{2} \ldots\ldots -2, -1, 0, 1, 2, \ldots\ldots \frac{N}{2} \tag{615}$$

- **Blackmann Window:** A data weighting window used prior to frequency transformation (FFT) providing improvements over the *Bartlett* and *Von Hann* windows by increasing spectral leakage rejection. For *N* data samples, the *Blackmann window* is defined by:

$$h(n) = \sum_{k=0}^{2} a(k)\cos\left(\frac{2kn\pi}{N}\right) \quad \text{for} \quad n = \frac{N}{2} \ldots\ldots\text{-2,-1,0,1,2} \ldots\ldots \frac{N}{2} \tag{616}$$

  with coefficients:

  $a(0)$ = 0.42659701, $a(1)$ = 0.49659062, $a(2)$ = 0.07684867

- **Blackmann-harris Window:** A type of data window often used in the calculation of FFTs/DFTs for reducing spectral leakage. Similar to the *Blackman* window, but with four cosine terms:

$$h(n) = \sum_{k=0}^{3} a(k)\cos\left(\frac{2kn\pi}{N}\right) \quad \text{for} \quad n = \frac{N}{2} \ldots\ldots\text{-2,-1,0,1,2} \ldots\ldots \frac{N}{2} \tag{617}$$

  with coefficients:

  $a(0)$ = 0.3635819, $a(1)$ = 0.4891775, $a(2)$ = 0.1365995, $a(3)$ = 0.0106411

- **Hamming Window:** A data weighting window used prior to frequency transformation (FFT) to reduce spectral leakage. Compared to the uniform window (no weighting) the *Bartlett window* doubles the width

of the main lobe, while attentuating the main sidelobe by 46dB, compared to the 13dB of the uniform window. Compared to the similar *Von Hann* window, the *Hamming window* sidelobes do not decay as rapidly. For *N* data samples, the Barlett window is defined by:

$$h(n) = 0.54 + 0.46\cos\left(\frac{2n\pi}{N}\right) \quad \text{for} \quad n = \frac{N}{2} \text{ .....-2,-1,0,1,2 .....} \frac{N}{2} \tag{618}$$

- **harris Window:** A data weighting window used prior to frequency transformation (FFT) to reduce spectral leakage (similar to the *Bartlett* and *Von Hann* windows). For *N* data samples, the *harris window* is defined by:

$$h(n) = \sum_{k=0}^{3} a(k)\cos\left(\frac{2kn\pi}{N}\right) \quad \text{for} \quad n = \frac{N}{2} \text{ .....-2,-1,0,1,2 .....} \frac{N}{2} \tag{619}$$

with coefficients:

a(0) = 0.3066923, *a(1)* = 0.4748398, *a(2)* = 0.1924696, *a(3)* = 0.0259983

- **Vonn Hann Window:** A data weighting window used prior to frequency transformation (FFT). Compared to the uniform window (no weighting) the *Von Hann* doubles the width of the main lobe, while attentuating the main sidelobe by 32dB, compared to the 13dB of the uniform window. For *N* data samples, the Von Hann window is defined by:

$$h(n) = 0.5 + 0.5\cos\left(\frac{2n\pi}{N}\right) \quad \text{for} \quad n = -\frac{N}{2} \text{ ..... -2 , -1 , 0, 1, 2, ...} \frac{N}{2} \tag{620}$$

Wold Decomposition: H. Wold showed that any stationary stochastic discrete time process, $x(n)$, can be decomposed into two components: (1) a general linear regression of white noise; and (2) a predictable process. The general linear regression of white noise is given by:

$$u(k) = 1 + \sum_{n=1}^{\infty} b_n v(k-n) \quad \text{with} \quad \sum_{n=1}^{\infty} |b_n| < \infty \tag{621}$$

and the predictable process, $s(n)$, can be entirely predicted from its own past samples. $s(n)$ and $v(n)$ are uncorrelated, i.e. $E\{v(n)s(k)\} = 0$ for all *n, k* [77]. See also *Autoregressive Modelling, Yule Walker Equations*.

Woodbury's Identity: See *Matrix Properties - Inversion Lemma.*

Wordlength: The size of the basic unit of arithmetic computation inside a DSP processor. For a fixed point DSP processor the wordlength is at least 16 bits, and in the case of the DSP56000, it is 24 bits. Floating point DSP processors usually use 32 bit wordlengths. See also *DSP Processor, Parallel Multiplier*.

World Wide Web (WWW): The *World Wide Web* (or *the web*) has become the de facto standard on the internet for storing, finding and transferring open information; hypertext (with text, graphics and audio) is used to access information. Most universities and companies involved in DSP now have *web* servers with home pages where the information available on a particular machine is summarised. There are also likely to be hypertext links available for cross referencing to additional information. The best way to understand the existence and usefulness of the World Wide Web is to use it with tools such as Mosaic or Netscape. Speak to your system manager or call up your phone company or internet service provider for more information.

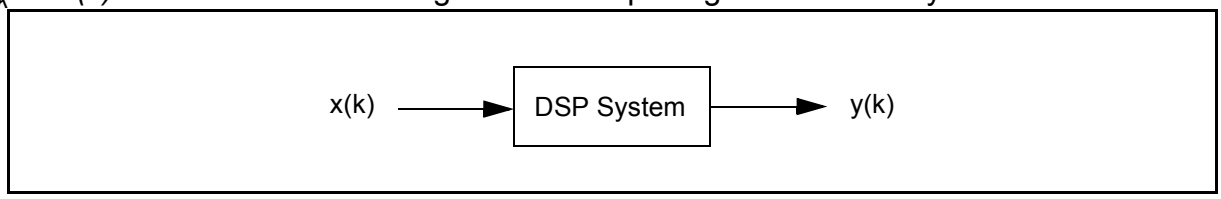
Woofer: The section of a loudspeaker that reproduces low frequencies is often called the *woofer*. The name is derived from the low pitched *woof* of a dog. The antithesis to the *woofer* is the tweeter. See also *Tweeter.*

X

X-Series Recommendations: The X-series telecommunication recommendations from the International Telecommunication Union (ITU), advisory committee on telecommunications (denoted ITU-T and formerly known as CCITT) provide standards for data networks and open system communication. For details on this series of recommendations consult the appropriate standard document or contact the ITU.

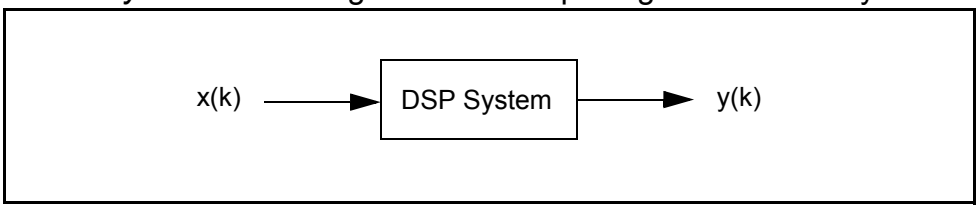
The well known X.400 standards are defined for the exchange of multimedia messages by store-and-forward transfer. The X.400 standards therefore provide an international service for the movement of electronic messages without restriction on the types of encoded information conveyed. The ITU formed a collaborative partnership with the International Organization for Standards for the development and continued definition of X.400 in 1988 (See ISO 10021 (Parts 1-7).) A joint technical committee was also formed by the ISO and the International Electrotechnical Commission (IEC). See also *International Electrotechnical Commission, International Organization for Standards, International Telecommunication Union, ITU-T Recommendations, Standards.*

x_k: x_k or *x(k)* is *often* the name assigned to the input signal of a DSP system.

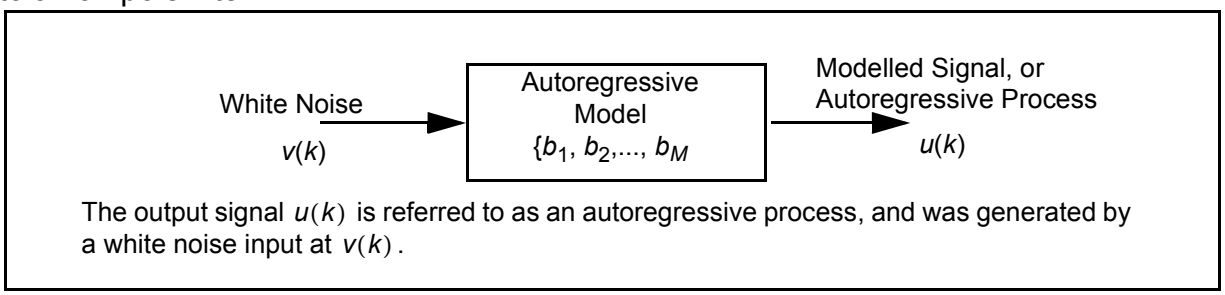


Y

y_k: y_k or *y(k)* is *usually* the name assigned to the output signal of a DSP system.



Yule Walker Equations: Consider a stochastic signal, $u(k)$ produced by inputting white noise, $v(k)$ to an all-pole filter:



The output signal $u(k)$ is referred to as an autoregressive process, and was generated by a white noise input at $v(k)$.

If the inverse problem is posed such that you are given the autoregressive signal $u(k)$ and the order of the process (say *M*), then the autoregressive filter weights $\{b_1, b_2, \ldots b_M\}$ that produced the given process from a white noise signal, $v(n)$ can be found by solving the Yule Walker equations:

$$\Rightarrow \boldsymbol{b}_{AR} = \boldsymbol{R}^{-1}\boldsymbol{r} \tag{622}$$

where the vector $\boldsymbol{b} = [b_1 \ \ldots \ b_{M-1} \ \ b_M]^T$, $\boldsymbol{R}$ is the $M \times M$ correlation matrix:

$$\boldsymbol{R} = E\{\boldsymbol{u}(k-1)\boldsymbol{u}^T(k-1)\} = \begin{bmatrix} r_0 & \ldots & r_{M-2} & r_{M-1} \\ \vdots & \ldots & \vdots & \vdots \\ r_{M-2} & \ldots & r_0 & r_1 \\ r_{M-1} & \ldots & r_1 & r_0 \end{bmatrix} \tag{623}$$

and $\boldsymbol{r}$ the $M \times 1$ correlation vector,

$$\boldsymbol{r} = E\{u(k)\boldsymbol{u}(k-1)\} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_M \end{bmatrix} \tag{624}$$

where $r_n = E\{u(k)u(k-n)\} = E\{u(k-n)u(k)\}$, where $E\{.\}$ is the expectation operator.

See also *Autoregressive Modelling*.

Z

Z^{-1}: Derived from the z-transform of signal, z^{-1} is taken to mean a delay of one sample period. Sometimes denoted simply as Δ.

Zeroes: A sampled impulse response (e.g. of a digital filter) can be transferred into the Z-domain, and the *zeroes* of the function can be found by factorizing the polynomial to find the roots:

$$H(z) = 1 - 3z^{-1} + 2z^{-1} = (1 - z^{-1})(1 - 2z^{-2}) \quad (625)$$

i.e. the zeros are $z = 1$ and $z = 2$.

Zero Order Hold: If a signal is upsampled or reconstructed by holding the same value until the next sample value, then this is a zero order hold. Also called step reconstruction. See *First Order Hold, Reconstruction Filter.*

Zero-Padding: See *Fast Fourier Transform - Zero Padding*.

Zoran: A manufacturer and designer of special purpose DSP devices.

Z-transform: A mathematical transformation used for theoretical analysis of discrete systems. Transforming a signal or a system into the z-domain can greatly facilitate the understanding of a particular system [10].

Common Numbers Associated with DSP

In this section numerical values which are in some way associated with DSP and its applications are listed. The entries are given in an alphabetical type order, where 0 is before 1, 1 is before 2 and so on, with no regard to the actual magnitude of the number. Decimal points are ignored.

0 dB: If a system attenuates a signal by 0 dB then the signal output power is the same as the signal input power, i.e.

$$10\log\frac{P_{\text{out}}}{P_{\text{in}}} = 10\log 1 = 0 \text{ dB} \tag{626}$$

0x: Used as a prefix by Texas Instruments processors to indicate hexadecimal numbers.

0.0250858... : The base 10 logarithm of the ratio of the fundamental frequency of any two neighboring notes (one semi-tone apart) on a musical instrument tuned to the Western music scale. See also *Western Music Scale*.

0.6366197: An approximation of $2/\pi$. See also *3.92dB*.

1 bit A/D: An alternative name for a Sigma-Delta (Σ-Δ) A/D.

1 bit D/A: An alternative name for a Sigma-Delta (Σ-Δ) D/A.

1 bit idea: An alternative name for a really stupid concept.

10^{-12} W/m^2: See entry for 2 x10-5 N/m2.

1004Hz: When measuring the bandwidth of a telephone line, the 0dB point is taken at 1004 Hz.

10149: The ISO/IEC standard number compact disc read only system description. Sometimes refered to as the Yellow Book. See also *Red Book.*

10198: The ISO/IEC standard number for JPEG compression.

1024: 2^{10}. The number of elements in 1k, when refering to memory sizes, i.e. 1 kbyte = 1024 bytes.

1.024 Mbits/sec: The bit rate of a digital audio system sampling at $f_s = 32000$ Hz with 2 (stereo) channels and 16 bits per sample.

1070 Hz: One of the FSK (frequency shift keying) carrier frequencies for the Bell 103, 300 bits/sec modem. Other frequencies are 1270 Hz, 2025 Hz and 2225 Hz.

103: The Bell 103 was a popular 300 bits/sec modem standard.

1.05946...: The twelfth root of 2, i.e $2^{1/12}$. This number is the basis of the modern western music scale whereby the ratio of the fundamental frequencies of any two adjacent notes on the scale is *1.05946*... See also *Music, Western Music Scale*.

10.8dB: Used in relation to quantisation noise power calculations; $10\log 1/12 = 10.8$ dB.

11.2896 MHz: 2×5.6448 MHz and used as a clock for oversampling sigma delta ADCs and DACs. 5.6448 MHz sampling frequency can be decimated by a factor of 128 to 44.1kHz ,a standard hifidelity audio sampling frequency for CD players.

115200 bits/sec: The 111520 bits/sec modem is an eight times speed version of the very popular 14400 modem and became available in the mid 1990s. This modem uses echo cancellation, data equalisation, and data compression technique to achieve this data rate. See also *300, 2400, V-series recommendations*.

11544: The ISO/IEC standard number for JBIG compression.

11172: The ISO/IEC standard number for MPEG-1 video compression.

120 dB SPL: The nominal threshold of pain from a sound expressed as a sound pressure level.

1200 Hz: The carrier frequency of the originating end of the ITU V22 modem standard. The answering end uses a carrier frequency of 2400Hz. Also one of the carrier frequencies for the FSK operation of the Bell 202 and 212 standards, the other one being 2400Hz.

1209 Hz: One of the frequency tones used for DTMF signalling. See also *Dual Tone Multi-frequency.*

12.288 MHz: 2×6.144 MHz and used as a clock for oversampling sigma delta ADCs and DACs. 6.144 MHz sampling frequency can be decimated by a factor of 128 to 48kHz, a standard hifidelity audio sampling frequency for DAT.

128: 2^7

12.8 MHz: 2×6.4 MHz and used as a clock for oversampling sigma delta ADCs and DACs. 6.4 MHz sampling frequency can be decimated by a factor of 64 to a sampling frequency of 100kHz.

13 dB: The attentuation of the first sidelobe of the function $10\log \sin x/x$ is approximately 13 dB. See also *Sine Function*.

1336 Hz: One of the frequency tones used for DTMF signalling. See also *Dual Tone Multi-frequency.*

13522: The ISO/IEC standard number for MHEG multimedia coding.

13818: The ISO/IEC standard number for MPEG-2 video compression.

-13 dB: The ISO/IEC standard number for MPEG-2 video compression.

1.4112 Mbits/sec: The bit rate of a CD player sampling at f_s = 44100Hz, with 2 (stereo) channels and 16 bits per sample.

14400 bits/sec: The 14400 bits/sec modems was six times speed version of the very popular 2400 modem and became available in the early 1990s, with the cost falling dramatically in a few years. See also *300, 2400, V-series recommendations*.

1.452 - 1.492 GHz: The 40 MHz radio frequency band allocated for satellite DAB (digital audio broadcasting) at the 1992 World Administrative Radio Conference in Spain. Due to other plans for this bandwidth, a number of countries selected other bandwidths such as 2.3 GHz in the USA, and 2.5 GHz in fifteen other countries.

147: The number of the European digital audio broadcasting (DAB) project started in 1987, and formally named Eureka 147. This system has been adopted by ETSI (the European

Telecommunication Standards Institute) for DAB and currently uses MPEG Audio Layer 2 for compression.

147:160: The largest (integer) common denominator of the sampling rates of a CD player, and a DAT player, i.e.

$$\frac{44100}{300} : \frac{48000}{300} = 147 : 160 \tag{627}$$

1477 Hz: One of the frequency tones used for DTMF signalling. See also *Dual Tone Multi-frequency.*

1.536 Mbits/sec: The bit rate of a DAT player sampling at f_s = 48000Hz, with 2 (stereo) channels and 16 bits per sample.

160: See *147*.

1633 Hz: One of the frequency tones used for DTMF signalling. See also *Dual Tone Multi-frequency.*

16384: 2^{14}

1.76 dB: Used in relation to quantisation noise power calculations; $10\log 1.5 = 1.76$ dB.

176.4kHz: The sample rate when 4 $\times$'s oversampling a CD signal where the sampling frequency $f_s = 44.1\text{kHz}$.

1800 Hz: The carrier frequency of the QAM (quadrature amplitude modelling) ITU V32 modem standard.

2 bits: American slang for a quarter (dollar).

2-D FFT: The extension of the (1-D) FFT into two dimensions to allow Fourier transforms on images.

2×10^{-5} N/m^2: The reference intensity, sometimes denoted as I_{ref}, for the measurement of sound pressure levels (SPL). This intensity can also be expressed as 10^{-12} W/m^2, or as $20\,\mu$Pa (micropascals). This intensity was chosen as it was close to the absolute level of a tone at 1000Hz that can just be detected by the human ear; the average human threshold of hearing at 1000Hz is about 6.5dB. The displacement of the eardrum at this sound power level is suggested to be 1/10th the diameter of a hydrogen molecule!

20 dB/octave: Usually used to indicate how good a low pass filter attenuates at frequencies above the 3dB point. 20dB per octave means that each time the frequency doubles then the attenuation of the filter increases by a factor of 10, since 20dB = $20\log_2(10)$. 20dB/decade is the same roll-off as 6dB/decade. See also *Decibels, Roll-off.*

20 μ Pa (micropascals): See entry for *2 x10-5 N/m2.*

205: The number of data points in used in Goertzel's algorithm (a form of discrete Fourier transform (DFT)) for tone detection.

2025 Hz: One of the FSK (frequency shift keying) carrier frequencies for the Bell 103, 300 bits/sec modem. Other frequencies are 1070 Hz, 1270 Hz and 2225 Hz.

2048: 2^{11}

2100: The part number of most Analog Devices fixed point DSP processors.

21000: The part number of most Analog Devices floating point DSP processors.

2225 Hz: One of the FSK (frequency shift keying) carrier frequencies for the Bell 103, 300 bits/sec modem. Other frequencies are 1070 Hz, 1270 Hz and 2025 Hz.

24 bits: The fixed point wordlength of some members of the Motorola DSP56000 family of DSP processors.

2400 bits/sec: The 2400 bits/sec modems appeared in the early 1990s as low cost communication devices for remote computer access and FAX transmission. The bit rate of 2400 was chosen as it is a factor of 8 faster than the previous 300 bits/sec modem. Data rates of 2400 were achieved by using echo cancellation and data equalisation techniques. The 2400 bits/sec modem dominated the market until the cost of the 9600 modems started to fall in about 1992. To ensure a simple backwards operation compatibility all modems are now produced in factors of 2400, i.e. *4800, 7200, 9600, 14400, 28800, 57600, 115200.* See also *V-series recommendations*.

2400 Hz: The carrier frequency of the answering end of the ITU V22 modem standard. The originating end uses a carrier frequency of 1200Hz. Also one of the carrier frequencies for the FSK operation of the Bell 202 and 212 standards, the other one being 1200Hz.

256: 2^8

26 dB: The attentuation of the first sidelobe of the function $20\log \sin x/x$ is approximately 26 dB. See also *Sine Function*.

261.624 Hz: The fundamental frequency of middle C on a piano tuned to the Western music scale. See also *440 Hz*.

2.718281... : The (truncated) value of *e*, the natural logarithm.

28800 bits/sec: The 28800 bits/sec modem is an eight times speed version of the very popular 14400 modem and became available in the mid 1990s. This modem uses echo cancellation, data equalisation, and data compression technique to achieve this data rate. See also *300, 2400, V-series recommendations*.

2.8224 MHz: An intermediate oversampling frequency used for sigma delta ADCs and DACs used with CD audio systems. 2.8224 MHz can be decimated by a factor of 64 to 44.1 kHz, the standard sampling frequency of CD players.

3 dB: See *3.01dB*.

3.01 dB: The approximate value of $10\log_{10}(0.5) = 3.0103$. If a signal is attenuated by 3dB then its power is halved.

300: The largest (integer) common denominator of the sampling rates of a CD player, and a DAT player, i.e.

$$\frac{44100}{300} : \frac{48000}{300} = 147 : 160 \tag{628}$$

300 bits/sec: The bit rate of the first commercial computer modems. Although 28800 bits/sec is now easily achievable, 300 bits/sec modems probably outsell all other speeds of modems by virtue of the fact that most credit card telephone verification systems can perform the verification task at 300 bits/sec in a few seconds. See also *Bell 103, 2400, V-series recommendations*.

3.072 MHz: An intermediate oversampling frequency used for sigma delta ADCs and DACs used with DAT and other professional audio systems. 3.072 MHz can be decimated by a factor of 64 to 48kHz, the current standard professional hifidelity audio sampling frequency.

32 kHz: A standard hifidelity audio sampling rate. The sampling rate of NICAM for terrestrial broadcasting of stereo audio for TV systems in the United Kingdom.

32 bits: The wordlength of most floating point DSP processors. 24 bits are used for the mantissa, and 8 bits for the exponent.

3.2 MHz: An intermediate oversampling frequency for sigma delta ADCs and DACs that can be decimated by a factor of 32 to 100 kHz.

320: The part number for most Texas Instruments DSP devices.

32768: 2^{15}

3.3 Volt Devices: DSP processor manufacturers are now releasing devices that will function with 3 volt power supplies, leading to a reduction of power consumption.

350 Hz: Tones at 350 Hz and 440 Hz make up the dialing tone for telephone systems.

35786 km: The height above the earth of a satellite geostationary orbit. This leads to between 240 and 270ms one way propagation delay for satellite enabled telephone calls. On a typical international telephone connection the round-trip delay can be as much as 0.6 seconds making voice conversation difficult. In the likely case of additional echoes voice conversation is almost impossible without the use of echo cancellation strategies.

+++ 352.8 bits/sec: One quarter of the bit rate of hifidelity CD audio sampled at 44.1 kHz, with 16 bit samples and stereo channels ($44100 \times 16 \times 2 = 1411200$ bits/sec). The data compression scheme known as PASC (psychoacoustic subband coding) used on DCC (digital compact cassette) compresses by a factor 4:1 and therefore has a data rate of *384 bits/sec* when used on data sampled at 44.1kHz.

& 352.8kHz: The sample rate when 8 $\times$'s oversampling a CD signal where the sampling frequency is $f_s = 44.1\text{kHz}$.

+++ 384 bits/sec: One quarter of the bit rate of hifidelity audio sampled at 48kHz, with 16 bit samples and stereo channels ($48000 \times 16 \times 2 = 1536000$ bits/sec). The data compression scheme known as PASC (psychoacoustic subband coding) used on DCC (digital compact cassette) compresses by a factor 4:1 and therefore has a data rate of *384 bits/sec* when used on data sampled at 44.1kHz.

& 3.92dB: The attenuation of the frequency response of a step reconstructed signal at $f_s/2$. The attenuation is the result of the zero order hold "step" reconstruction which is equivalent to

convolving the signal with a unit pulse of time duration $t_s = 1/f_s$, or in the frequency domain, multiplying by the sinc function, $H(f)$ ::

$$H(f) = \frac{\sin \pi f t_s}{\pi f t_s} \tag{629}$$

Therefore at $f_s/2$, the droop in the output signal spectrum has a value of:

$$H(f_s/2) = \frac{\sin(\pi/2)}{\pi/2} = \frac{2}{\pi} = 0.63662 \tag{630}$$

which in dB's can be expressed as:

$$20\log(2/\pi) = 3.922398 \tag{631}$$

4 dB: Sometimes used as an approximation to 3.92dB. See also *3.92dB*

4096: 2^{12}

4294967296: 2^{32}

440 Hz: The fundamental frequency of the first A note above middle C on a piano tuned to the Western music scale. Definition of the frequency of this one note allows the fundamental tuning frequency of all other notes to be defined.

Also the pair of tones at 440 Hz and 350 Hz make up the telephone dialing tone, and 440 Hz and 480 Hz make up the ringing tone for telephone systems.

44.1kHz: The sampling rate of Compact Disc (CD) players. This sampling frequency was originally chosen to be compatible with U-matic video tape machines which had either a 25 or 30Hz frame rate, i.e. 25 and 30 are both factors of 44100.

44.056kHz: The sampling rate of Compact Disc (CD) players. was originally chosen to be compatible with U-matic video tape machines which had either a 25 or 30Hz frame rate, i.e. 25 and 30 are both factors of 44100. When master recording was done on a 29.97Hz frame rate video machine, this required the sampling rate to be modified to a nearby number that was a factor of 29.97, i.e. 44.056kHz. This sampling rate is redundant now.

4.76cm/s: The tape speed of compact cassette players, and also of digital compact cassette players (DCC).

4.77 dB: $10\log 3 \approx 4.77\text{dB}$, i.e. a signal that has its power amplfied by a factor of 3, has an amplification of 4.77dB.

48kHz: The sampling rate of digital audio tape (DAT) recorders, and the sampling rate used by most professional audio systems.

480 Hz: The tone pair 480 Hz and 620 Hz make up the busy signal on telephone systems.

4800 bits/sec: The 4800 bits/sec modems was a double speed version of the very popular 2400 modem. Data rates of 4800 were achieved using echo cancellation and data equalisation techniques. See also *2400, V-series recommendations*.

512: 2^9

56000: The part number for most Motorola fixed point DSP devices.

5.6448 MHz: An oversampling frequency for sigma delta ADCs and DACs used with CD players. 5.6448 MHz can be decimated by a factor of 128 to 44.1kHz the standard hifidelity audio sampling frequency for CD players.

57200 bits/sec: The 57200 bits/sec data rate modem is an 4 times speed version of the very popular 14400 modem and became available in the mid 1990s. This modem uses echo cancellation, data equalisation, and data compression technique to achieve this data rate. See also *300, 2400, V-series recommendations*.

6dB/octave: The "6" is an approximation for $20\log_{10}2 = 6.0206$. Usually used to indicate how good a low pass filter attenuates at frequencies above the 3dB point. 6dB per octave means that each time the frequency doubles then the attenuation of the filter increases by a factor of 2, since. 6dB/octave is the same roll-off as 20dB/decade. See also *Decibels, Roll-off.*

6.144 MHz: An oversampling frequency for sigma delta ADCs and DACs used with DAT and other professional audio systems. 6.144 MHz can be decimated by a factor of 128 to 48kHz to the current standard professional hifidelity audio sampling frequency.

620 Hz: The tone pair 480 Hz and 620 Hz make up the busy signal on telephone systems.

6.4 MHz: An oversampling frequency for sigma delta ADCs and DACs that can be decimated by a factor of 64 to 100 kHz.

64kBits/sec: A *standard* channel bandwidth for data communications. If a channel has a bandwidth of approximately 4kHz, then the Nyquist sampling rate would be 8kHz, and data of 8 bit wordlength is sufficient to allow good fidelity of speech to be transmitted. Note that 64000 bits/sec = 8000Hz × 8 bits.

6.4 MHz: A common sampling rate for a 64 times oversampled sigma-delta (Σ-Δ) A/D, resulting in up to 16 or more bits of resolution at 100kHz after decimation by 64.

65536: 2^{16}

697 Hz: One of the frequency tones used for DTMF signalling. See also Dual Tone Multi-frequency.

& 705600 bits/sec: The bit rate of a single channel of a CD player, with 16 bit samples, and sampling at $f_s = 44100\text{kHz}$.

& 705.6 kHz: The sample rate when 16 $\times$'s oversampling a CD signal where the sampling frequency $f_s = 44100\text{kHz}$.

7200 bits/sec: The 7200 bits/sec modems was a three times speed version of the very popular 2400 modem and became available in the early 1990s, with the cost falling dramatically in a few

years. Data rates of 7200 were achieved using echo cancellation and data equalisation techniques. See also *2400, V-series recommendations*.

741 Op-Amp: The part number of a very popular operational amplifier chip widely used for signal conditioning, amplification, and anti-alias, reconstruction filters.

768000 bits/sec: The bit rate of a single channel DAT player with 16 bits per sample, and sampling at $f_s = 48000$ Hz.

770 Hz: One of the frequency tones used for DTMF signalling. See also *Dual Tone Multi-frequency.*

8 kHz: The sampling rate of most telephonic based speech communication.

8192: 2^{13}

852 Hz: One of the frequency tones used for DTMF signalling. See also *Dual Tone Multi-frequency.*

941 Hz: One of the frequency tones used for DTMF signalling. See also *Dual Tone Multi-frequency.*

9.54dB: $20\log 3 \approx 9.54\text{dB}$, i.e. a signal that has its voltage amplfied by a factor of 3, has an amplification of 9.54 dB.

9600 bits/sec: The 9600 bits/sec modems was a four times speed version of the very popular 2400 modem and became available in the early 1990s, with the cost falling dramatically in a few years. Data rates of 9600 were achieved by using echo cancellation and data equalisation techniques. See also *2400, V-series recommendations*.

96000: The part number for most Motorola 32 bit floating point devices.

Acronyms:

ADC - Analogue to Digital Converter.

ADSL - Advanced Digital Subscriber Line

ADSR - Attack-Decay-Sustain-Release.

AES/EBU - Audio Engineering Society/European Broadcast Union.

A/D - Analogue to Digital Converter.

ADPCM - Adaptive Differential Pulse Code Modulation.

ANC - Active noise cancellation.

ANSI - American National Standards Institute.

AIC - Analogue Interfacing Chip.

ARB - Arbitrary Waveform Generation.

ASCII - American Standard Code for Information Interchange.

ASIC - Application Specific Integrated Circuit.

ASK - Amplitude Shift Keying.

ASPEC - Adaptive Spectral Perceptual Entropy Coding .

ASSP - Acoustics, Speech and Signal Processing.

AVT - Active Vibration Control.

AWGN - Additive White Gausssian Noise.

BER - Bit Error Rate.

BISDN - Broadband Integrated Services Digital Network.

BPF - Band pass filter.

BPSK - Binary Phase Shift Keying.

CCR - Condition Code Register.

CCITT - Comité Consultatif International Télégraphique et Téléphonique. (International Consultative Committee on Telegraphy and Telecommunication, now known as ITU-T.)

CCIR - Comité Consultatif International Radiocommunication. (International Consultative Committee on Radiocommunication, now known as ITU-R.)

CD - Compact Disc

CD-DV: Compact Disc Digital Video.

CELP - Coded Excited Linear Prediction Vocoders.

CENELEC - Comité Européen de Normalisation Electrotechnique (European Committee for Electrotechnical Standardization).

CIF - Common Intermediate Format.

CIRC - Cross Interleaved Reed Solomon code.

CISC - Complex Instruction Set Computer.

CPM - Continuous Phase Modulation.

CPU - Central Processing Unit.

CQFP - Ceramic Quad Flat Pack.

CRC - Cyclic Redundancy Check.

CVSD - Continuous variable slope delta modulator.

D/A - Digital to analogue converter.

DAB - Digital Audio Broadcasting.

DAC - Digital to analogue converter.

dB - decibels.

DECT - Digital European Cordless Telephone.

DL - Difference Limen.

& DARS - Digital Audio Radio Services.

DBS - Direct Broadcast Satellites.

DCC - Digital Compact Cassette.

DCT - Discrete Cosine Transform.

& DDS - Direct Digital Synthesis.

DECT - Digital European Cordless Telephone.

DFT - Discrete Fourier Transform.

DLL - Dynamic Link Library.

DMS - Direct Memory Access.

DPCM - Differential Pulse Code Modulation.

DPSK - Differential Phase Shift Keying.

DRAM - Dynamic Random Acces Memory.

DSL - Digital Subscriber Line

DSP - Digital Signal Processing.

DTMF - Dual tone Multifrequency.

DSfP - Digital Soundfield Processing.

ECG - Electrocardiograph.

EEG - Electroencephalograph.

EFM - Eight to Fourteen Modulation.

EMC - Electromagnetic compatibility.

EPROM - Electrically programmable read only memory.

EEPROM - Electrically Erasable Programmable Read Only Memory.

EQ - Equalization (usually in acoustic applications).

ETSI - European Telecommunications Standards Institute.

FIR - Finite Impulse Response.

FFT - Fast Fourier Transform.

FSK - Frequency Shift Keying.

G - prefix meaning 10^9**, as in GHz, thousands of millions of Hertz GII -** Global Information Infrastructure.

GIF - Graphic Interchange Format.

GSM - Global System For Mobile Communications (Group Speciale Mobile).

HDSL - High speed Digital Subscriber Line

hhtp - Hypertext Transfer Protocol.

IEEE - Institute of Electrical and Electronic Engineers (USA).

IEE - Institute of Electrical Engineers (UK).

IEC - International Electrotechnical Commission.

IIR - Infinite impulse response.

IIF - Image Interchange Facility.

INMARSAT - International Mobile Satellite Organization.

ISDN - Integrated Services Digital Network.

ISO - International Organisation for Standards.

ISO/IEC JTC - International Organization for Standards/ International Electrotechnical Commission Joint Technical Committee.

ITU - International Telecommunications Union.

ITU-R - International Telecommunications Union - Radiocommunication.

ITU-T - International Telecommunications Union - Telecommunication.

I/O - Input/Output.

JBIG - Joint Binary Image Group.

JND - Just Noticeable Difference.

JPEG - Joint Photographic Expert Group.

JTC - Joint Technical Committee.

k - prefix meaning 10^3, as in kHz, thousands of Hertz.

LFSR - Linear Feedback Shift Register Coding.

LPC - Linear Predictive Coding.

LSB - Least Significant Bit.

M - prefix meaning 10^6 as in MHz, millions of Hertz.

MAC - Multiply Accumulate.

MFLOPS - Millions of Floating Point Operations per Second.

MIDI - Music...

MAF - Minimum Audible Field.

MAP - Minimum Audible Pressure.

MIPS - Millions of Instructions per second.

MLPC - Multipulse Linear Predictive Coding.

MA - Moving Average.

MD - Mini-Disc.

MMSE - Minimum Mean Squared Error.

MHEG - Multimedia and Hypermedia Experts Group.

MPEG - Moving Picture Experts Group.

MRELP - M..

ms - millisecond (10^{-3}).

MSB - Most Significant Bit.

MSE - Mean Squared Error.

MSK - Minimum Shift Keying.

MIX - Modular Interface eXtension.

MUSICAM - Masking pattern adapted Universal Subband Integrated Coding And Multiplexing.

NRZ - Non Return to Zero.

ns - nanosecond (10^{-9} seconds).

OKPSK - Offset-Keyed Phase Shift Keying.

OKQAM - Offset-Keyed Quadrature Amplitude Modulation.

OOK - On Off Keying.

OPSK - Offset-Keyed Phase Shift Keying.

OQAM - Offset-Keyed Quadrature Amplitude Modulation.

PAM - Pulse Amplitude Modulation.

PASC - Precision Adaptive Subband Coding.

PCM - Pulse Code Modulation.

PCMCIA - Personal Computer Memory Card International Association.

PN - Pseudo-Noise.

ppm - Parts per million.

PPM - Pulse Position Modulation.

PRBS - Pseudo Random Binary Sequence.

PSK - Phase Shift Keying.

PSTN - Public Switched Telephone Network.

PTS - Permanent Threshold Shift.

PWM - Pulse Width Modulation.

PDA - Personal Digital Assistant.

PGA - Pin Grid Array.

PID - Proportional Integral Controller.

PQFP - Plastic Quad Flat Pack.

PRNS - Pseudo Random Noise Sequence.

QAM - Quadrature Amplitude Modulation.

QPSK - Quadrature Phase Shift Keying.

RAM - Random access memory.

RBDS - Radio Broadcasting.....?

RELP - Residual Excited Linear Prediction Vocoder.

RIFF - Resource Interchange File Format.

RISC - Reduced Instruction Set Computer.

RLC - Run Length Coding.

RLE - Run Length Encoding.

ROM - Read only memory.

RPE - Recursive Predictor Error **or** Regular Pulse Excitation

RZ - Return to Zero.

Rx - Receive.

SBM - Super Bit Mapping (A trademark of Sony).

SCMS - Serial Copy Management System.

SFG - Signal Flow Graph.

SGML - Standard Generalized Markup Language.

S/H - Sample and Hold.

SINR - Signal to Interference plus Noise Ratio.

SNR - Signal to Noise Ratio.

S/N - Signal to Noise ratio.

S/P-DIF - Sony/Philips Digital Interface Format.

SR - Status Register.

SPL - Sound Pressure Level.

SRAM - Static random access memory.

SRC - Sample Rate Converter.

TBDF - Triangular Probability Density Function.

TCM - Trellis Coded Modulation.

THD - Total Harmonic Distortion.

THD+N - Total Harmonic Distortion plus Noise.

TTS - Temporary Threshold Shift.

Tx - Transmit.

VSELP - Vector Sum Excited Linear Prediction.

VU - Volume Unit.

WMA - Weighted Moving Average.

WWW - World Wide Web.

μ **sec -** microsecond (10^{-6})

Standards Organisation

ANSI - American National Standards Institute.

BS - British Standard.

IEC - International Electrotechnical Committee.

IEEE - Institute of Electronic and Electrical Engineers.

ISO - International Organisation for Standards.

References and Further Reading

Goto Papers

Textbooks

[1] S. Banks. Signal Processing, Image Processing and Pattern Recognition. Prentice Hall, Englewood Cliffs, NJ, 1990.

[2] T.P. Barnwell III, K. Nayebi, C.H. Richardson. Speech Coding. A Computer Laboratory Textbook. John Wiley and Sons, 1996.

[3] A. Bateman and W. Yates. Digital Signal Processing Design. Pitman Publishing 1988.

[4] E.H. Berger, W.D. Ward, J.C. Morrill, L.H. Royster. Noise and Hearing Conservation Manual, 4th Edition. American Industrial Hygiene Association.

[5] R.L. Brewster. ISDN Technology. Chapman & Hall, London, 1993.

[6] R.G. Brown, P.Y.C. Hwang. Introduction to Random Signals and Applied Kalman Filtering, John Wiley and Sons, 1992.

[7] C.S. Burrus, J.H. McLellan, A.V. Oppenheim, T.W. Parks, R.W. Schafer, H.W. Schuessler. Computer Based Exercises for Signal Processing Using Matlab. Prentice Hall, 1994.

[8] J.C. Candy, G.C. Temes. Oversampling Delta-Sigma Data Converters. Piscataway, NJ; IEEE Press, 1992.

[9] L.W. Couch II. Modern Communication Systems: Principles and Applications. Prentice-Hall, Englewood Cliffs, NJ, 1995.

[10] D.J. DeFatta, J.G. Lucas, W.S. Hodgkiss. Digital Signal Processing: A System Design Approach. John Wiley, New York, 1988.

[11] J.R. Deller, J.G. Proakis, J.H.K. Hansen. Discrete Time Processing of Speech Signals. MacMillan, New York, 1993.

[12] P.D. Denyer and D. Renshaw. VLSI Signal Processing - A Bit Serial Approach. Addison-Wesley, 1995.

[13] G. De Poli, A Piccialli, C. Roads. Representations of Musical Signals. The MIT Press, Boston, USA, 1991.

[14] J.M. Eargle. Music Sound and Technology. Van Nostrand Reinhold, 1990.

[15] G.H. Golub, C.F. Van Loan. Matrix Computations. John Hopkins University Press, 1989.

[16] J.G. Gibson, The Mobile Communications Handbook. CRC Press/IEEE Press, 1996.

[17] S. Haykin. Adaptive Filter Theory (2nd Edition). Prentice Hall, Englewood Cliffs, NJ, 1990.

[18] S. Haykin. Neural Networks: A Comprehensive Foundation. MacMillan College, 1994.

[19] D.R. Hush and B.G. Horne. Progress in supervisied neural networks. IEEE Signal Processing Magazine, Vol. 10, No. 1, pp. 8-39, January 1993.

[20] K. Hwang, F. Briggs. Computer Architecture and Parallel Processing. McGraw-Hill, 1985.

[21] E.C. Ifeachor, B.W. Jervis. Digital Signal Processing: A Practical Approach. Addison-Wesley, 1993.

[22] N. Kalouptsidis,Theodoridis. Adaptive System Identification and Signal Processing Algorithms. Prentice Hall, 1993.

[23] A. Kamas, E.A. Lee. Digital Signal Processing Experiments. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[24] S.Y. Kung. Digital Neurocomputing. Prentice-Hall, Englewood Cliffs, NJ, 1992.

[25] S.Y. Kung. VLSI Array Processors. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[26] P.A. Lynn. An Introduction to the Analysis and Processing of Signals, 1982.

[27] J.D. Martin. Signals and Processes: A Foundation Course.

[28] C. Marven, G. Ewers. A Simple Approach to Digital Signal Processing. Texas Instruments Publication, 1993.

[29] R.M. Mersereau, M.J.T. Smith. Digital Filtering. John Wiley, New York, 1993.

[30] B.C.J Moore. An Introduction to the Psychology of Hearing.

[31] A.V. Oppenheim, R.W. Schafer. Discrete Time Signal Processing. Prentice Hall, Englewood Cliffs, NJ, 1989.

[32] R.A. Penfold. Synthesizers for Musicians. PC Publishing, London, 1989.

[33] K. Pohlmann. Advanced Digital Audio. Howards Sams, Indiana, 1991.

[34] K. Pohlmann. An Intoduction to Digital Audio, Howard Sams, Indiana, 1989.

[35] T.S. Rappaport. Wireless Communications. IEEE Press, New York, 1996.

[36] P. Regalia. Adaptive IIR Filtering. Marcel Dekker, 1995.

[37] F. Rumsey. Digital Audio. Butterworth-Heinemann, 1991

[38] E. Rogers and Y. Li. Parallel Processing in a Control Systems Environment. Prentice Hall, Englewood Cliffs, NJ, 1993.

[39] K. Sayood. Introduction to Data Compression. Morgan-Kaufman, 1995.

[40] M. Schwartz. Information, Transmission, and Modulation Noise. McGraw-Hill.

[41] N.J.A. Sloane, A.D. Wyner (Editors). Claude Elwood Shannon: Collected Papers. IEEE Press, 1993, Piscataway, NJ. ISBN 0-7803-0434-9.

[42] M.J.T. Smith, R.M Mersereau. Introduction to Digital Signal Processing: A Computer Laboratory Textbook. John Wiley and Sons, 1992.

[43] K. Steiglitz. A Digital Signal Processing Primer. Addison-Wesley, 1996.

[44] C.A. Stewart and R. Atkinson. Basic Analogue Computer Techniques. McGraw-Hill, London, 1967.

[45] N. Storey. Electronics: A Systems Approach. Addison-Welsey, 1992.

[46] M. Talbot-Smith (Editor). Audio Engineer's Reference Book, Focal Press, ISBN 0 7506 0386 0, 1994.

[47] F.J. Taylor. Principles of Signals and Systems. New York; McGraw-Hill, 1994.

[48] W.J. Tompkins. Biomedical Digital Signal Processing. Prentice Hall, Englewood Cliffs, NJ, 1993.

[49] P.P. Vaidyanathan. Multirate Systems and Filter Banks. Prentice Hall, Englewood Cliffs, NJ,1993.

[50] S.V. Vaseghi. Advanced Signal Processing and Digital Noise Reduction. John Wiley/B.G. Tuebner, 1996.

[51] J. Watkinson. An Introduction to Digital Audio. Focal Press, ISBN 0 240 51378 9, 1994.

[52] J. Watkinson. Compression in Video and Audio. ISBN 0240513940, Focal Press, April 1994.

[53] B. Widrow and S. Stearns. Adaptive Signal Processing. Prentice Hall, 1985.

[54] J. Watkinson. The Art of Digital Audio, 2nd Edition. ISBN 0240 51320 7, 1993.

Technical Papers

[55] P.M. Aziz, H.V. Sorenson, J.V. Der Spiegel. An overview of sigma delta converters. IEEE Signal Processing Magazine, Vol. 13, No. 1, pp. 61-84, January 1996.

[56] J.W. Arthur. Modern SAW-based pulse compression systems for radar application. Part 2: Practical systems. IEE Electronics Communication Engineering Journal, Vol. 8, No. 2, pp. 57-78, April 1996.

[57] G.M. Blair. A Review of the Discrete Fourier Transform. Part 1: Manipulating the power of two. IEE Electronics and Communication Engineering Journal, Vol. 7, No.4, pp. 169-176, August 1995.

[58] G.M. Blair. A review of the discrete Fourier Transform. Part 2: Non-radix algorithms, real transforms and noise. IEE Electronics Communication Engineering Journal, Vol. 7, No. 5, pp. 187-194, October 1995.

[59] J.A. Cadzow. Blind deconvolution via cumulant extrema. IEEE Signal Processing Magazine, Vol. 13, No. 3, pp. 24-42, May 1996.

[60] J. Cadzow. Signal processing via least squares error modelling. IEEE ASSP Magazine, Vol. 7, No. 4, pp 12-31, October 1990.

[61] G. Cain, A. Yardim, D. Morling. All-Thru DSP Provision, Essential for the modern EE. Proceedings of IEEE International Conference on Acoutics, Speech and Signal Processing 93, pp. I-4 to I-9, Minneapolis, 1993.

[62] C. Cellier. Lossless audio data compression for real time applications. 95th AES Convention, New York, Preprint 3780, October 1993.

[63] S. Chand and S.L. Chiu (editors). Special Issue on Fuzzy Logic with Engineering Applications. Proceedings of the IEEE, Vol. 83, No. 3, pp. 343-483, March 1995.

[64] R. Chellappa, C.L. Wilson and S. Sirohey. Human and machine recognition of faces. Proceedings of the IEEE, Vol. 83, No. 5 pp. 705-740, May 1995.

[65] J. Crowcroft. The Internet: a tutorial. IEE Electronics Communication Engineering Journal, Vol. 8, No. 3, pp. 113-122, June 1996.

[66] J.W. Cooley. How the FFT gained acceptance. IEEE Signal Processing Magazine, Vol. 9, No. 1, pp. 10-13, January 1992.

[67] J.R. Deller, Jr. Tom, Dick and Mary discover the DFT. IEEE Signal Processing Magazine, Vol. 11, No. 2, pp. 36-50, April 1994.

[68] S.J. Elliot and P.A. Nelson. Active Noise Control. IEEE Signal Processing Magazine, Vol. 10, No. 4, pp 12-35, October 1993.

[69] L.J. Eriksson. Development of the filtered-U algorithm for active noise control. Journal of the Acoustical Society of America, Vol. 89 (No.1), pp 27-265, 1991.

[70] H. Fan. A (new) Ohio yankee in King Gustav's country. IEE Signal Processing Magazine, Vol.12, No. 2, pp. 38-40, March 1995.

[71] P.L. Feintuch. An adaptive recursive LMS filter. Proceedings of the IEEE, Vol. 64, No. 11, pp. 1622-1624, November 1976.

[72] D. Fisher. Coding MPEG1 image data on compact discs. Electronic Product Design (UK), Vol. 14, No. 11, pp. 26-33. November 1993.

[73] H. Fletcher and W.A. Munson. Loudness, its definition, measurement and calculation. Journal of the Acoustical Society of America, Vol. 70, pp. 1646-1654, 1933.

[74] M. Fontaine and D.G. Smith. Bandwidth allocation and connection admission control in ATM networks. IEE Electronics Communication Engineering Journal, Vol. 8, No. 4, pp. 156-164, August 1996.

[75] W. Gardner. Exploitation of spectral redundancy in cyclostationary signals. IEEE Signal Processing Magazine, Vol. 8, No. 2, pp 14-36, April 1991.

[76]H. Gish and M. Schmidt. Text independent speaker identification. Vol. 11, No. 4, pp 18- 32, October 1994.

[77] P.M. Grant. Signal processing hardware and software. IEEE Signal Processing Magazine, Vol. 13, No. 1, pp. 86-88, January 1996.

[78] S. Harris. The effects of sampling clock jitter on Nyquist sampling analog to digital converters, and on oversampling delta sigma ADCs. Journal of the Audio Engineering Society, July 1990.

[79] S. Heath. Multimedia standards and interoperability. Electronic Product Design (UK), Vol. 15, No. 9, pp. 33-37. November 1993.

[80] F. Hlawatsch and G.F. Boudreaux-Bartels. Linear and quadratic time-frequency signal representations. IEEE Signal Processing Magazine, Vol. 9, No. 2, pp. 21-67, April 1992.

[81]D.R. Hush and B.G. Horne. Progress in Supervised Neural Networks. IEEE Signal Processing Magazine, Vol. 10, No. 1, pp. 8-39, January 1993.

[82] Special Issue on DSP Education, IEEE Signal Processing Magazine, Vol. 9, No.4, October 1992.

[83] Special Issue on Fuzzy Logic with Engineering Applications. Proceedings of the IEEE, Vol. 83, No. 3, March 1995.

[84] A. Hoogendoorn. Digital Compact Cassette. Proceedings of the IEEE, Vol. 82, No. 10, pp. 1479-1489, October 1994.

[85] B. Jabbari (editor). Special Issue on Wireless Networks for Mobile and Personal Communications. Vol. 82, No. 9, September 1994.

[86] D.L. Jaggard (editor). Special Section on Fractals in Electrical Engineering. Proceedings of the IEEE, Vol. 81, No. 10, pp. 1423-1523, October 1993.

[87] N. Jayant, J. Johnston, R. Safranek. Signal compression based on models of human perception. Proceedings of the IEEE, Vol. 81, No. 10, pp. 1385-1382, October 1993.

[88] C.R. Johnson. Yet still more on the interaction of adaptive filtering, identification, and control. IEE Signal Processing Magazine, Vol.12, No. 2, pp. 22-37, March 1995.

[89] R.K. Jurgen. Broadcasting with Digital Audio. IEEE Spectrum, Vol. 33, No. 3, pp. 52-59. March 1996.

[90] S.M. Kay and S.L. Marple. Spectrum Analysis - A Modern Perspective. Proceedings of the IEEE, Vol. 69, No. 11, pp 1380-1419, November 1981.

[91] K. Karnofsky. Speeding DSP algorithm design. IEEE Spectrum, Vol. 33, No. 7, pp. 79-82, July 1996.

[92] W. Klippel. Compensation for non-linear distortion of horn loudspeakers by digital signal processing. Journal of the Audio Engineering Society, Vol. 44, No. 11, pp 964-972, Novemeber 1996.

[93] P. Kraniauskas. A plain man's guide to the FFT. IEEE Signal Processing Magazine, Vol. 11, No. 2, pp. 24-36, April 1994.

[94] F. Kretz and F. Cola. Standardizing Hypermedia Information Objects. IEEE Communications Magazine, May 1992.

[95] M. Kunt (Editor). Special Issue on Digital Television, Part 1: Technologies. Proceedings of the IEEE, Vol. 83, No. 6, June 1995.

[96] M. Kunt (Editor). Special Issue on Digital Television, Part 2: Hardware and Applciations. Proceedings of the IEEE, Vol. 83, No. 7, July 1995.

[97] T.I. Laakso, V. Valimaki, M. Karjalainen and U.K. Laine. Splitting the unit delay. IEEE Signal Processing Magazine, Vol. 13, No. 1, pp. 30-60, January 1996.

[98] T.I. Laasko, V. Valimaki, M. Karjalainen, U.K. Laine. Splitting the Unit Delay. IEEE Signal Processing Magazine, Vol. 13, No. 1, pp. 30-60, January 1996.

[99] P. Lapsley and G. Blalock. How to estimate DSP processor perfomance. IEEE Spectrum, Vol. 33, No. 7, pp. 74-78, July 1996.

[100]V.O.K. Li and X. Qui. Personal communication systems. Proceedings of the IEEE, Vol. 83, No. 9, pp. 1210-1243, September 1995.

[101]R.P. Lippmann. An introduction to computing with neural nets. IEEE ASSP Magazine, Vol. 4, No. 2, pp. 4-22, April 1987.

[102]G.C.P. Lokhoff. DCC: Digital Compact Cassette. IEEE Transactions on Consumer Electronics, Vol. 37, No. 3 pp 702-706, August 1991.

[103]H. Lou. Implementing the Viterbi Algorithm. IEEE Signal Processing Magazine. Vol 12, No. 5 pp. 42-52, September 1995.

[104].J. Lipoff. Personal communications networks bridging the gap between cellular and cordless phones. Proceedings of the IEEE, Vol. 82, No. 4, pp. 564-571, April 1994.

[105]M. Liou. Overview of the p*64 kbit/s Video Coding Standard. Communications of the ACM, April 1991. G.K. Ma and F.J. Taylor. Multiplier policies for digital signal processing. IEEE ASSP Magazine, Vol. 7, No. 4, pp 6-20, January 1990.

[106]G-K. Ma and F.J. Taylor. Multiplier policies for digital signal processing. IEEE ASSP Magazine, Vol. 7, No. 1, January 1990.

[107]Y. Mahieux, G. Le Tourneur and A. Saliou. A microphone array for multimedia workstations. Journal of the Audio Engineering Society, Vol. 44, No. 5, pp. 331-353, May 1996.

[108]D.T. Magill, F.D. Natali and G.P. Edwards. Spread-spectrum technology for commercial applications. Proceedings of the IEEE, Vol. 82, No. 4, pp. 572-584, April 1994.

[109]V.J. Mathews. Adaptive Polynomial Filters. IEEE Signal Processing Magazine, Vol. 8, No. 3, pp. 10-26, July 1991.

[110]N. Morgan and H. Bourland. Neural networks for statistical recognition of continuous speech. Proceedings of the IEEE, Vol. 83, No. 5, pp. 742-770, May 1995.

[111]N. Morgan and H. Bourland. Continuous speech recognition. IEEE Signal Processing Magazine, Vol. 12, No. 3, pp. 24-42, May 1995.

[112]N. Morgan and H. Bourland. Neural networks for statistical recognition of speech. Proceedings of the IEEE, Vol 83, No. 5, pp 742-770, May 1995.

[113]A. Miller. From here to ATM. IEEE Spectrum, Vol. 31, No. 6, pp 20-24, June 1994.

[114]Y.K. Muthusamy, E. Barnard and R.A. Cole. IEEE Signal Processing Magazine, Vol. 11, No. 4, pp. 33-41, October 1994.

[115]R.N. Mutagi. Psuedo noise sequences for engineers. IEE Electronics Communication Engineering Journal, Vol.

8, No. 2, pp. 79-87, April 1996.

[116]R.N. Mutagi. Pseudo noise sequences for engineergs. IEE Electronics and Communication Engineering Journal, Vol. 8. No. 2, pp. 79-87, April 1996.

[117]C.L. Nikias and J.M Mendel. Signal processing with higher order statistics. IEEE Signal Processing Magazine, Vol. 10, No. 3, p 10-37, July 1993.

[118]P.A. Nelson, F. Orduna-Bustamante, D. Engler, H. Hamada. Experiments on a system for the synthesis of virtual acoustic sources. Journal of the Audio Engineering Society, Vol. 44, No. 11, pp 973-989, Novemeber 1996.

[119]P.A. Nelson, F. Orduna-Bustamante, H. Hamada. Multichannel signal processing techniques in the reproduction of sound. Journal of the Audio Engineering Society, Vol. 44, No. 11, pp 973-989, Novemeber 1996.

[120]P. Noll. Digital audio coding for visual communications. Proceedings of the IEEE, Vol. 83, No. 6, pp. 9 925-943, June 1995

[121]K.J. Olejniczak and G.T. Heydt. (editors). Special Section on the Hartley Transform. Proceedings of the IEEE, Vol. 82, No . 3, pp. 372-447, March 1994.

[122]J. Picone. Continuous speech recognition using hidden Markov models. IEEE ASSP Magazine, Vol. 7, No. 3, pp. 26-41, July 1990.

[123]M. Poletti. The design of encoding functions for stereophonic and polyphonic sound systems. Journal of the Audio Engineering Society, Vol. 44, No. 11, pp 948-963, November 1996.

[124]P.A. Ramsdale. The development of personal communications. IEE Electronics Communication Engineering Journal, Vol. 8, No. 3, pp. 143-151, June 1996.

[125]P. Regalia, S.K. Mitra, P.P. Vaidynathan. The digital all-pass filter: a versatile building block. Proceedings of the IEEE, Vol. 76, No. 1, pp. 19-37, January 1988.

[126]D.W. Robinson and R.S. Dadson. A redetermination of the equal loudness relations for pure tones. British Journal of Applied Physics, Vol. 7, pp. 166-181, 1956.

[127]R.W. Robinson. Tools for Embedded Digital Signal Processing. IEEE Spectrum, Vol. 29, No. 11, pp 81-84, November 1992.

[128]C.W. Sanchez. An Understanding and Implementation of the SCMS Serial Copy Management System for Digital Audio Transmission. 94th AES Convention, Preprint #3518, March 1993. R. Schafer and T. Sikora. Digital video coding standards and their role in video communications. Proceedings of the IEEE, Vol. 83, No. 6, pp. 907-924, June 1995.

[129]C.E. Shannon. A mathematical theory of Communication. The Bell System Technical Journal, Vol. 27, pp. 379-423, July 1948. (Reprinted in Claude Elwood Shannon: Collected Papers [41].)

[130]C.E. Shannon. The Bandwagon (Editorial). Institute of Radio Engineers, Transations on Information Theory, Vol. IT-2, p. 3 March 1956. (Reprinted in Claude Elwood Shannon: Collected Papers [41].)

[131]J.J. Shynk. Frequency domain and multirate adaptive filtering. IEEE Signal Processing Magazine, Vol. 9, No. 1, pp. 10-37, January 1992.

[132]J.J. Shynk. Adaptive IIR filtering. IEEE ASSP Magazine, Vol. 6, No. 2, pp. 4-21, April 1989.

[133]H.F. Silverman and D.P. Morgan. The appliation of dynamic programming to converted speech recognition. IEEE ASSP Magazine, Vol. 7, No. 3, pp. 6-25, July 1990.

[134]J.L. Smith. Data compression and perceived quality. IEEE Signal Processing Magazine, Vol. 12, No. 5, pp. 58-59, September 1995.

[135]A.S. Spanias. Speech coding: a tutorial review. Proceedings of the IEEE, Vol. 82, No. 10, pp. 1541-1582, October 1994.

[136]A.O. Steinhardt. Householder transforms in signal processing. IEEE Signal Processing Magazine, Vol. 5, No. 3, pp. 4-12, July 1988.

[137]R.W. Stewart. Practical DSP for Scientist. Proceedings of IEEE International Conference on Acoutics, Speech and Signal Processing 93, pp. I-32 to I-35, Minneapolis, 1993.

[138]C.Stone. Infrasound. Audio Media, Issue 55, AM Publishing Ltd, London,June 1995.

[139]J.A. Storer. Special Section on Data Compression. Proceedings of the IEEE, Vol. 82, No. 6, pp. 856-955, June 1994.

[140]JP. Strobach. New forms of Levinson and Schur algorithms. IEEE Signal Processing Magazine, Vol. 8, No. 1, pp. 12-36, January 1991.

[141].R. Treicher, I. Fijalkow, and C.R. Johnson, Jr. Fractionally spaced equalizers. IEEE Signal Processing Magazine, Vol. 13, No. 3, pp. 65-81, May 1996.

[142]B.D.Van Veen and K. Buckley. Beamforming: A Versatile Approach to spatial filtering. IEEE ASSP Magazine, Vol. 5, No.2, pp. 4-24, April 1988.

[143]V. Valimaki, J. Huopaniemi, M. Karjalainen and Z. Janosy. Physical modeling of plucked string instruments with application to real time sound synthesis. Journal of the Audio Engineering Society, Vol. 44, No. 5, pp. 331-353, May 1996.

[144]V.D. Vaughn and T.S. Wilkinson. System considerations for multispectral image compression designs. IEEE Signal Processing Magazine, Vol. 12, No. 1, pp. 19-31, January 1995.

[145]S.A. White. Applications of distributed arithmetic to digital signal processing: a tutorial review.

[146]IEEE ASSP Magazine, Vol. 6, No. 3, pp. 4-19, July 1989.

[147]W.H.W. Tuttlebee. Cordless telephones and cellular radios: synergies of DECT and GSM. IEE Electronics Communication Engineering Journal, Vol. 8, No. 5, pp. 213-223, October 1996.

[148]Working Group on Communication Aids for the Hearing Impaired. Speech perception aids for hearing impaired people: current status and needed research. Journal of Acoustical Society of America, Vol. 90, No.2, 1991

[149]R.D. Wright. Signal processing hearing aids. Hearing Aid Audiology Group, Special Publication, British Society of Audiology, London, 1992.

[150]F. Wylie. Digital audio data compression. IEE Electronics and Communication Engineering Journal, pp. 5-10, February 1995.

[151]I. Wickelgren. The Strange Senses of Other Species. IEEE Spectrum, Vol. 33, No. 3, pp. 32-37. March 1996.

[152]B. Widrow et al. Adaptive Noise Cancellation: Principles and Applications. Proceedings of the IEEE, Vol. 63, pp. 1692-1716, 1975.

[153]B. Widrow et al. Stationary and Non-stationary learning characteristics of the LMS adaptive filter. Proc. IEEE, Vol 64, pp. 1151-1162, 1976.

[154]T. Yamamoto, K. Koguchi, M. Tsuchida. Proposal of a 96kHz sampling digital audio. 97th AES Convention, October 1994, Audio Engineering Society preprint 3884 (F-5).

[155]T. Yoshida. The rewritable minidisc system. Proceedings of the IEEE, Vol. 82, No. 10, pp. 1492-1500 October 1994.

[156]Y.Q. Zhang, W. Li, M.L. Liou (Editors). Special Issue on Advances in Image and Video Compression. Proceedings

of the IEEE, Vol. 83, No. 2, February 1995.

[157]British Society of Audiology. Recommended procedures for pure tone audiometry. British Journal of Audiometry, Vol. 15, pp213-216, 1981.

[158]IEC-958/ IEC-85, Digital Audio Interface / Amendment. International Electrotechnical Commission, 1990.

[159]DSP Education Session. Proceedings of IEEE International Conference on Acoutics, Speech and Signal Processing 92, pp. 73-109, San Francisco, 1992.

[160]Special Section on the Hartley Transform (Edited by K.J. Olejniczak and G.T. Heydt). Proceedings of the IEEE, Vol. 82, No. 3, March 1994.

[161]Special Issue on Advances in Image and Video Compression (Edited by Y.Q. Zhang, W. Li and M.L. Liou). Proceedings of the IEEE, Vol. 83, No. 2, February 1995.

[162]Special Issue on Digital Television Part 2: Hardware and Applications (Editor M. Kunt). Proceedings of the IEEE, Vol. 83, No. 7, July 1995.

[163]Special Issue on Electrical Therapy of Cardiac Arrhythmias (Edited by R.E. Ideker and R.C. Barr). Proceedings of the IEEE, Vol. 84, No. 3, March 1996.

[164]Special Section on Data Compression (Editor J.A. Storer). Proceedings of the IEEE, Vol. 82, No. 6, June 1994.

[165]Special Section on Field Programmable Gate Arrays (Editor A. El Gamal). Proceedings of the IEEE, Vol. 81, No. 7, July 1993.

[166]Special Issue on Wireless Networks for Mobile and Personal Communications (Editor B. Jabbari). Proceedings of the IEEE, Vol. 82, No. 9, September 1994.

[167]Special Issue on Digital Television, Part 1: Technologies (Editor M. Kunt). Proceedings of the IEEE, Vol. 83, No. 6, June 1995.

[168]Special Issue on Time-Frequency Analysis (Editor P.J. Loughlin). Proceedings of the IEEE, Vol. 84, No. 9, September 1996.

[169]Technology 1995. IEEE Spectrum, Vol. 32, No.1, January 1995.